

AD-A178 400

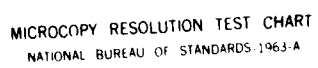
A DESCRIPTIVE EVALUATION OF AUTOMATED SOFTWARE  
COST-ESTIMATION MODELS(U) INSTITUTE FOR DEFENSE  
ANALYSES ALEXANDRIA VA E K BAILEY ET AL. OCT 86  
IDA-P-1979 MDA903-84-C-0031

1/2

UNCLASSIFIED

F/G 9/2

ML



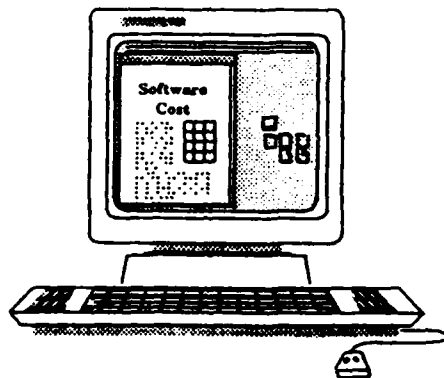
MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A



# A Descriptive Evaluation of Automated Software Cost-Estimation Models

AD-A178 400

DTIC



DTIC  
SELECTE  
MAR 25 1987  
S D

DTIC FILE COPY

**DISTRIBUTION STATEMENT A**

Approved for public release;  
Distribution Unlimited

Published by:

The Office of Assistant Secretary of Defense (Comptroller) and the  
Institute for Defense Analyses (IDA Paper P-1979)

Dated: October 1986

87 3 12 032

IDA PAPER P-1979

# **A DESCRIPTIVE EVALUATION OF AUTOMATED SOFTWARE COST-ESTIMATION MODELS**

Elizabeth K. Bailey  
Thomas P. Frazier  
John W. Bailey

October 1986



**INSTITUTE FOR DEFENSE ANALYSES**

Contract MDA 903-84-C-0031  
Task Order T-5-326



COMPTROLLER

ASSISTANT SECRETARY OF DEFENSE

WASHINGTON, DC 20301-1100

The Department's general purpose ADP systems are absolutely essential in supporting our everyday business and operational functions. Today, and in the future, improved information systems will be integral to solving many of our most critical defense problems. However, aging computer systems are also sometimes part of the problem.

Many of our large information systems were originally developed in the 70's and are just now in the process of being replaced. Consequently, we will, in the near-term, experience increased costs associated with the development of modern, functionally efficient, applications software. Effective management control of these costs is essential if we are to continue to improve our systems processes and thus ultimately achieve a more efficiently operating Department of Defense.

I believe that the management concepts which are presented in this volume, and the type of products which are described, provide a real opportunity to exert significant management control over the cost and schedule aspects of application software development and maintenance.

I urge you to use these types of tools to improve the management of DoD software. If you have any questions or comments regarding this document, please address them to my Directorate for Information Resources Management Systems.

*Robert W. Hill*



Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By <i>lth on file</i>	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

## FOREWORD

→ The purpose of this report is to encourage and facilitate the use of appropriate software cost and schedule models in the Department of Defense. This report provides an introduction to the usefulness of these models and an evaluation of current, fully supported, commercially available, software cost and schedule estimation products.

These products can be employed for analyses of several different aspects of software management including:

- Project planning - to arrive at an early estimate of resource requirements and identify tradeoffs between cost and schedule;
- Proposal evaluation - to assess the realism of contract proposals and assist in negotiation;
- Managing development - to track resources, get early warning of deviation from plan, and assess alternatives for corrective action.
- Managing maintenance - to estimate and track the cost of maintenance and enhancement and identify appropriate timing for a new start;
- Improving software productivity - to evaluate methodologies and tools which may increase productivity.

An important advantage of these parametric models is that they make visible the many assumptions about the product and the development organization that enter into an estimate. These assumptions are therefore open to discussion, validation, and assessment of alternatives. This visibility provides a basis for assessing risk, evaluating reasonableness of a plan, and taking management action or negotiating a bid. In summary, software cost and schedule models form a basis for sharpening software management skills.

Directorate for Information Resources Management Systems

## ACKNOWLEDGMENTS

The authors wish to express their gratitude to a number of persons who have helped in this project. Mr. Rob Cooper of the Office of the Assistant Secretary of Defense - Comptroller has served as the project officer and provided thoughtful and consistent guidance throughout the effort. Several experts in the field of software cost-estimating models have given generous amounts of their time in discussion over the past few months. Among those, we especially would like to mention; Captain Joe Dean (USAF), Mr. Troy Cox and Mr. Jon Shimer of the National Security Agency; Ms. Mary Dempsey, GTE-Eastern Division; Mr. Andrew Najberg, The Analytic Sciences Corporation; Mr. Steve Gross, Department of the Navy; Mr. Paul Garvey, The MITRE Corporation; and Mr. Angelo Barrone, Department of the Army. Dr. Steve Balut, Mr. Paul Goree and Mr. Robert Simmons of the Cost Analysis Group and Ms. Sarah Nash and Dr. John Salasin of the Computer Software and Engineering Division read drafts of this report and provided helpful advice. Ms. Julie Dossett provided her usual excellent manuscript preparation support.

## CONTENTS

LETTER OF TRANSMITTAL.....	iii
FOREWORD.....	v
ACKNOWLEDGMENTS.....	vii
I. INTRODUCTION.....	1
A. BACKGROUND.....	1
B. OBJECTIVES AND LIMITATIONS.....	3
C. REPORT OUTLINE.....	4
II. EVALUATION CRITERIA .....	5
A. USAGE CATEGORY #1.....	6
B. USAGE CATEGORY #2.....	6
C. USAGE CATEGORY #3.....	8
D. USAGE CATEGORY #4.....	10
E. USAGE CATEGORY #5.....	11
III. SELECTION OF MODELS AND GENERAL DESCRIPTION...	13
A. SELECTION CRITERIA.....	13
B. OVERVIEW OF THE COST MODELS.....	14
C. VENDORS.....	15
D. HARDWARE CONFIGURATION.....	17
E. CONTRACTUAL ARRANGEMENTS.....	17
F. COSTS.....	18
G. LIFE-CYCLE RANGE COVERED.....	19
H. INPUT PARAMETERS.....	19
I. QUALITY OF THE USER DOCUMENTATION.....	23
IV. EVALUATION OF MODELS.....	25
V. SURVEYS OF USERS AND POTENTIAL USERS OF COST MODELS .....	31
A. SURVEY 1: Use of Cost Models in MCCR Programs.....	31
B. SURVEY 2: Former Participants in OASD-Sponsored Training.....	33
C. SURVEY 3: Known Users of Cost Models.....	35
VI. FINDINGS.....	37



## CONTENTS (continued)

### APPENDICES

A. REFERENCES.....	A-1
B. GENERAL INFORMATION.....	B-1
C. MODEL INPUT PARAMETERS.....	C-1
D. DETAILED EVALUATION OF MODELS.....	D-1

### EXHIBITS

Exhibit 1. MODEL VENDORS.....	16
Exhibit 2. HARDWARE REQUIREMENTS.....	17
Exhibit 3. CONTRACTUAL ARRANGEMENTS.....	18
Exhibit 4. COSTS (DoD Rates).....	18
Exhibit 5. LIFE-CYCLE RANGE COVERED.....	19
Exhibit 6. DISTRIBUTION OF INPUT PARAMETERS BY INPUT CATEGORIES.....	21
Exhibit 7. PROPORTION OF OVERLAP AMONG PAIRS OF MODELS.....	22
Exhibit 8. QUALITY OF USER DOCUMENTATION.....	23
Exhibit 9. CRITERIA FOR USAGE CATEGORY #1 .....	26
Exhibit 10. CRITERIA FOR USAGE CATEGORY #2 .....	27
Exhibit 11. CRITERIA FOR USAGE CATEGORY #3 .....	28
Exhibit 12. CRITERIA FOR USAGE CATEGORY #4.....	29
Exhibit 13. CRITERIA FOR USAGE CATEGORY #5.....	30
Exhibit 14. REASONS FOR NOT USING MODELS .....	32

## **I. INTRODUCTION**

### **A. BACKGROUND**

One of the critical problems facing the U.S. Department of Defense (DoD) in the next decade is managing the rapid growth in software. The Electronic Industries Association [1] estimates that DoD Mission Critical Computer Resource (MCCR) expenditures for software currently exceed \$11 billion and will triple to over \$35 billion by 1995. If one were to add to these figures the expenditures for general purpose computer software (inventory control, payroll, etc.) that will be incurred over the next decade, the totals are an ever increasing percentage of the entire DoD budget.

Dollar figures provide only one indication of the increasing importance of software to the DoD. The growing sophistication and intelligence of defense systems is the direct result of computer technology and, in particular, of the software which controls the computers. This software is extremely complex and becoming more so. At the same time, however, it must be highly reliable and fault tolerant. The DoD faces the challenge of significantly increasing productivity while improving quality for software which continues to increase in size and complexity.

The difficulty of affecting significant advances is evident when one considers that the software acquisition process has been characterized by large cost overruns and schedule slippages in many programs. Thus, an additional challenge is to improve the ability to manage (predict and track) software resources consisting of people, dollars and time. The current report is concerned with this challenge and, in particular, with the contribution that formal (parametric) cost models can make toward improving the predictability of software projects. It should be noted that cost models can indirectly contribute to improving productivity and quality by providing information needed to make cost-effective decisions regarding improvements in the software process.

Software cost models typically take the form of a set of base equations which relate size, effort, and calendar time and, hence, allow the prediction of effort and time given size as an input parameter. The effort and time predictions are then modified by a number of additional parameters which reflect conditions specific to the project and organization of

interest (for example, complexity of the software, capability of personnel, use of tools). There are other ways of estimating software resource requirements. Boehm [2] discusses the use of expert opinion, reasoning-by-analogy in which estimates are based on results from similar projects, and the process of constructing estimates in a bottom-up manner.

Most parametric cost models embody aspects of one or more of these alternative methods. For example, in the COCOMO model presented in [2], the weightings given to the various input parameters were derived in part from expert opinion (and in part from historical data). An important advantage of parametric cost models is that they make visible the many assumptions about the product and the development organization that enter into a resource estimate. These assumptions are therefore open to discussion and debate. This visibility provides a basis for assessing the risk involved in an estimate, for evaluating the reasonableness of a bid, and for negotiating a contract.

Cost models can be employed for several kinds of analyses at different points in time. These include during:

- the early planning stage to arrive at an order-of-magnitude estimate of resource requirements
- the proposal evaluation process to assess the validity of contractor proposals
- the development activity to track on-going resource expenditures and to generate cost-to-complete estimates
- the operational phase to estimate and track the cost of maintenance and enhancements
- any phase to assist in identifying steps for increasing software productivity and quality.

The Office of the Assistant Secretary of Defense (Comptroller) (OASD(C)) has historically promoted the use of software cost models in the DoD. As part of this program, OASD(C) obtained a DoD-wide license for the use of a commercial automated cost model in the early 1980's. In addition to the licensing agreement, several hundred DoD personnel attended a three-day course on the use of the model. Although this model was, and continues to be, held in high regard by the software cost-estimating community, the course and licensing did not lead to widespread use by DoD components. The recent expiration of this license provided an opportunity for DoD to explore a variety of options for continuing the program including:

- a new contract similar to the one just completed

- licenses with several vendors who have introduced automated cost models since the original contract was awarded
- a policy allowing DoD components to select and use models of their own choosing.

## **B. OBJECTIVES AND LIMITATIONS**

The purpose of this report is to present the results of work performed as part of an ongoing IDA study conducted for DoD. Specifically, this report presents details of IDA's efforts to

- identify and evaluate the currently available software cost-models that could be used by components of the Department of Defense, and
- survey the DoD software community to assess the current state of practice with regard to the use of software cost models.

A large number of cost models have been proposed in the literature. A survey conducted by IDA identified a significant number of cost models [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. A number of these models are complex in terms of their underlying formulas, requiring a good deal of mathematical sophistication on the part of the user. In addition, the effort to predict resource requirements quickly becomes cumbersome on a project of any size when one is left with only a calculator and paper and pencil as tools. Thus, the widespread use of cost models will require the use of automated cost-estimation packages which are based on a specific cost model but which remove the burden of calculation from the user. For this reason, the evaluation of specific models which was performed as part of the first objective was limited to those which are automated and commercially available.

All of the models described estimate personnel effort, calendar time and staffing levels. Several estimate defects and operational reliability as well. The term "cost model" is a conventional one but it should be recognized that these are, in reality, cost-schedule-(and sometimes)-quality estimation models.

It should be noted that the criteria presented in this report say nothing about the accuracy of the estimates produced by the models. Empirical comparisons [20] have shown that, within a given software organization, there are large differences in the accuracy of the predictions made by different models. There are also wide fluctuations in the accuracy of any given cost model across different software organizations. Empirical

comparisons are necessary to determine which model is most accurate for a given organization.

### **C. REPORT OUTLINE**

This report is divided into six chapters and four appendices. Following this chapter, a set of criteria which provide a common framework for describing and evaluating commercially available and supported cost-estimation models are presented in Chapter II. Chapter III presents a description of how the models were selected and some general information about the set chosen for evaluation. Chapter IV provides a summary of the descriptive evaluation of these models. Chapter V details the survey of the DoD software community. The purpose of this survey was to determine the frequency and nature of cost model use within DoD components. Chapter VI summarizes the findings of the model evaluations and the survey. Appendix A presents a list of works cited in the body of the report. Appendix B details general information (e.g., costs, hardware configurations required) about each of the models examined in this report. Appendix C presents a detailed listing of the user inputs required to run the models. Appendix D presents the full text supporting the summary tables contained in Chapter IV.

## II. EVALUATION CRITERIA

This chapter delineates a set of criteria which provide a common framework for describing and evaluating cost models. The criteria are organized according to their relevance (from a DoD perspective) to five general uses for cost models. These uses include:

- (1) assistance in making basic investment decisions early in the life cycle,
- (2) assistance in validating contractor proposals,
- (3) support for day-to-day project management,
- (4) assistance in predicting enhancement and repair activities during operations and maintenance, and
- (5) support for analyses to identify major cost drivers and needed productivity improvements.

For each of these different uses, a somewhat different set of capabilities is desirable. For example, the ability to provide cost and schedule estimates on the basis of minimal input is important for supporting early investment decisions. On the other hand, for day-to-day project planning and tracking, a cost model should require a user to think about a project in much more detail.

The criteria are organized according to their relevance to the five uses outlined above. The criteria are not intended to identify the one "best" model in any absolute sense. Rather, they are intended to help users choose an appropriate cost model (or models) for their specific needs. By understanding which criteria are relevant for which use, and how alternative models measure up against the criteria (in a qualitative, descriptive sense), users are provided a reasonable basis for choosing a model. A brief description of each usage category precedes a list of the criteria which are relevant to that category.

**A. USAGE CATEGORY #1: Assistance in making basic investment decisions early in the life cycle**

Before major investment decisions can be made about a project, information is needed concerning the project's basic parameters. A number of questions must be answered including the following:

- What is the size of the software?
- How much should it cost?
- How long will it take?
- How many people will be required?
- How confident can we be in these estimates?

All current cost models provide, at a minimum, estimates of effort, calendar time, and staffing level. A cost model which is suitable for use early in the software life cycle must be able to provide this information with minimal inputs. Given that there are many unknowns at this early point in time, the estimate will necessarily reflect a high degree of uncertainty. The model should provide some measure of this uncertainty or, in some way, make clear the rough nature of the estimates. The size of the software to be developed is a major cost driver in all cost models yet size is one of the most difficult input parameters to estimate accurately. A model should provide help to the user in deriving a size estimate. In general, a cost model should provide sufficient information to allow for a "go/no-go" decision and to allow for the initial planning of project resources. The specific criteria for this category are presented below.

- 1.1 Can the model provide estimates on the basis of minimal input?
- 1.2 Does the model provide an indication of the uncertainty or risk associated with estimates?
- 1.3 Is assistance provided in estimating project size?

**B. USAGE CATEGORY #2: Assistance in validating contractor proposals**

Cost models can provide a valuable source of input in assessing the validity of costs and schedules contained in contractor proposals. Questions to be answered in making this assessment include the following:

- Are the proposed costs reasonable?
- Are the proposed schedules reasonable?

- How did the contractor arrive at these estimates, that is, what assumptions are being made about the software product and about the contractor's own organization?
- Are the cost and schedule estimates consistent with what the contractor has done in the past?
- How likely is the contractor to complete the software within the proposed cost and schedule?

From the perspective of a program office, a cost model can remove much of the mystery from contractor proposals and, thus, can provide a concrete basis for discussion and negotiation. In this regard, an important property of cost models is their replicability - that is, given the same set of input values and the same cost model, two independent organizations (that is, contractor and government) should arrive at the same estimates. The input values reflect assumptions about the to-be-developed product and the environment in which it will be developed. Thus, any variance in the two sets of estimates reflects differences in these assumptions which can be the subject of negotiation between the two parties. In order to facilitate this type of negotiation, a cost model should have easily understood input parameters for which variations have a readily interpretable meaning.

A difficult aspect of this type of assessment is that it must be carried out across a variety of development environments. A cost model should take these differences into account and should allow for a direct comparison between organizations in terms of their general capability to develop software.

Another difficulty in attempting to compare vendor proposals is that, in the absence of specific cost or schedule constraints, the estimates are likely to differ in terms of both of these parameters. One contractor, for example, may propose a relatively low cost but with an extended delivery date while another may propose a much higher cost along with a shorter schedule. A third contractor may propose both a low cost and shortened schedule, leading one to question whether this is feasible at all. A cost model should help in analyzing these cost-schedule tradeoffs and in gauging the feasibility of the estimates. Thus, it should indicate the likelihood of completing the software within the proposed cost and schedule.

Since there are cases in which the delivery date is fixed and/or a ceiling on costs has been imposed, a cost model should allow the user to impose one or more constraints and then determine if a feasible solution exists.



A cost model should handle not only newly-developed software but the adaptation of existing software as well. Finally, it should help the estimator strive for completeness by including costs other than those related to direct labor, such as computer resources, travel and living expenses, and inflation. The criteria for this category are presented below.

- 2.1 Does the model have easily understood input parameters for which different values have a readily interpretable meaning?
- 2.2 Can the model be calibrated to different environments?
- 2.3 Does the model support a comparison between two or more organizations in terms of capability?
- 2.4 Does the model support an analysis of cost-schedule tradeoffs?
- 2.5 Does the model indicate the likelihood of completing the software within the proposed cost and schedule?
- 2.6 Does the model allow developmental constraints to be imposed?
- 2.7 Is previously developed software handled?
- 2.8 Does the model include costs other than labor?
- 2.9 Does the model account for inflation?

**C. USAGE CATEGORY #3: Support for day-to-day project management**

One of the primary benefits of using a cost model is that it provides a framework for project planning and for tracking expenditures and progress against the plan. Questions of interest in project planning include the following:

- What is the work breakdown structure, that is, how is the total job decomposed into phases, milestones, and activities?
- How do the activities map into team and individual work assignments?
- What sequential dependencies exist among activities?
- Which activities lie on a critical path?
- How many people and how much calendar time should be allocated to individual software components?
- What is the monthly cash flow?
- What are the monthly staffing requirements?
- What is the effect of changes in work assignments, in personnel capability, or in some other factor?

Questions of interest in project tracking include:

- Is the project on track in terms of dollars expended, milestones completed, and product quality?
- What is the cost to complete?

In order to serve as an effective planning tool, much more is needed than the prediction of effort, cost, and schedule for the software system as a whole. The cost model must allow the user to decompose the software being estimated into individual components and to map the values being estimated onto phases and activities, providing what amounts to a work breakdown structure. The model should provide an analysis of the dependencies among the various activities. In particular, it should help in identifying those that lie on a critical path in which any delay in the completion of a critical-path activity will delay the scheduled delivery date. In contrast, non-critical-path activities can often be extended over a greater period of calendar time, allowing for flexibility in allocating personnel without adversely affecting the project schedule. By pointing to these kinds of dependencies, a cost model can go a long way toward helping a project manager develop a plan which optimizes the particular set of goals and constraints for the project.

The model should also provide support for sensitivity ("what-if") analyses allowing managers to derive answers to questions such as the following: "If I add three more people to the development of this component, what impact will that have on costs and on the project schedule?", "If I assign programmers of average rather than very high capability, what will be the impact?"

To facilitate day-to-day tracking, the estimates provided by a cost model should be broken down so they are presented at frequent intervals (weekly or monthly). The model should allow for easy comparison between planned and actual expenditures and milestone-completion dates as well as easy updating of input parameters so that revised estimates can be made. The end result should be that surprise schedule slippages and cost overruns are minimized.

Since much of project management involves the preparation of status reports and briefings, report generation capabilities - especially in the form of graphics - are important. The criteria for this category are presented below.

- 3.1 Does the model allow the user to decompose the software system into smaller components?
- 3.2 Does the model provide the equivalent of a work breakdown structure?

- 3.3 Does the model support the analysis of task dependencies?
- 3.4 Does the model provide support for sensitivity ("what-if") analyses?
- 3.5 Does the model provide a staffing plan that is broken down into frequent intervals, such as monthly?
- 3.6 Does the model provide a cash-flow plan?
- 3.7 Does the model provide a comparison of planned expenditures and milestone-completion dates versus actuals?
- 3.8 Does the model provide graphical capabilities?

**D. USAGE CATEGORY #4:** Assistance in predicting enhancement and repair activities during operations and maintenance

Within most software organizations, maintenance activities have very low visibility. This lack of visibility is reflected in the area of cost-modeling as well. In spite of the fact that maintenance costs outweigh development costs, most models concentrate on development. Questions of interest during the operational phase include:

- How many people are needed to maintain the software?
- How much will a specific enhancement (or upgrade or optimization) cost and how long will it take to implement?
- Is it cheaper in the long run to rebuild this component from scratch rather than continue trying to maintain it?

In order to support the operations and maintenance phase, a cost model should estimate the cost and time required for such activities as enhancements, optimizations, upgrades to new hardware and defect corrections. These estimates should take characteristics of the product into account rather than being calculated solely as a proportion of development costs. An often difficult decision that must be made by a maintenance organization is whether it is more cost-effective to develop a component from scratch than to modify or continue repairing an existing component. A cost model should provide help in making this decision.

The quality of the software is a major factor in determining maintenance costs. A cost model should provide an estimate of quality using either a static measure such as the number of defects at delivery and/or a measure of operational performance such as the mean time to failure. The criteria for this category are presented below.

4.1 Does the model provide post-delivery estimates of the operational phase, including optimizations, enhancements, defect corrections?

4.2 Does the model predict the number or density of defects?

4.3 Does the model predict the run-time behavior of the software using a measure such as mean-time-to-failure (MTTF)?

**E. USAGE CATEGORY #5: Support for analyses to identify major cost drivers and needed productivity improvements**

Another major benefit of using a cost model is that it provides a framework for systematic data collection for productivity analyses. Many software organizations cannot answer basic questions about their own activities such as:

- How accurate have our cost and schedule projections been?
- In what areas have we been most accurate versus most in error?
- What is our current productivity?
- Is this level of productivity part of an increasing trend?
- What are our major cost drivers?
- What can we do to control these?

By keeping an historical record of the initial input parameters and estimates as well as the updated values over time, one has the foundation for a valuable database to guide future management decisions. These data can be used to increase the accuracy of estimation for future projects, to record changes in productivity or quality over time, and to determine areas needing further improvement. They can also help an organization avoid making the same mistakes twice.

To support these types of project analyses, a cost model must store the values of input parameters, the resulting estimates and updated values. It should support analyses within a project as well as comparisons across projects. To improve the accuracy of future estimates, it should be possible for the user to calibrate the model using local data.

An important determinant of whether or not a cost model assists in these types of analyses is the extent to which it is a "white box", meaning that its algorithms and parameter weights are made available to and changeable by the user, or a "black box", meaning that the user can only observe inputs and outputs with the internals remaining hidden. White box models can be adapted by the experienced user to increase the accuracy of their predictions. The criteria for this category are presented below.

- 5.1 Does the model maintain a database of input values?
- 5.2 Does the model maintain a database of the resulting estimates?
- 5.3 Does the model maintain a database of actual values?
- 5.4 Does the model maintain a multiple-project database which can be usefully accessed by the user to provide comparisons and views of improvements over time?
- 5.5 Does the model accept earlier data for the purposes of calibration or comparison?
- 5.6 Can the model be characterized as a "white box"?

### **III. SELECTION OF MODELS AND GENERAL DESCRIPTION**

#### **A. SELECTION CRITERIA**

A large number of cost models have been proposed in the software engineering literature. A literature search conducted by the Institute for Defense Analyses yielded approximately twenty different cost models which have been referenced since the late 1960's. (These references appear in Appendix A.) While there is a large pool of potential models to choose from, there is a big difference between a model which exists only on paper and a model which exists in the form of an automated tool. For one thing, a paper model typically requires far more mathematical sophistication on the part of the users. In addition, a paper model may quickly become cumbersome for a project of any size. A total of seven commercially-available cost-estimation packages which are intended for general use were identified and selected for evaluation. Several additional models with these characteristics have subsequently been identified. Future updates of this evaluation will include these and others as they appear in the marketplace. The seven products included in the current evaluation are:

- JS-3 (Version 1.03D)
- PCOC (Version 7.01)
- PRICE S
- SLIM (Version 1.1)
- SoftCost (Version 5.1)
- SPQR/20 (Version 1.1)
- WICOMO (Version 1.3)

These seven are referred to as cost "models" but it should be borne in mind that they might more properly be called "tools" or "packages" (or even more properly "cost/schedule/quality estimation tools"). This and the following chapter are focused on this set of seven.

The present chapter contains general information about each of the cost models. This information is intended to be of value to potential users regardless of how they plan to use a given model. The chapter begins with a brief overview of each of the models, pointing to noteworthy features. This overview is followed by information about the

vendors, contractual arrangements involved in obtaining a license for each model, the costs involved, and the portion of the software life cycle addressed by each. Appendix B contains additional details about these areas and information about milestones defined, about cost elements encompassed by the estimates, and about the size of projects handled by each model and any additional constraints. The present chapter also examines the models in terms of their input parameters, with particular interest in the extent to which there is agreement across models on the classes of inputs required. The chapter ends with a brief look at the quality of the user documentation for the seven models.

The evaluations, which are summarized in Chapter IV, are based almost entirely on information contained in the user's manual for each system [21, 22, 23, 24, 25, 26, 27]. Additional information, particularly regarding the cost and contractual arrangements, was obtained from telephone conversations with the vendors.

It should be noted that most of these models are undergoing continual revision and in several cases considerably more sophisticated versions are under development. The individual vendors should be contacted for more information.

## **B. OVERVIEW OF THE COST MODELS**

JS-3 and SLIM share many of the same assumptions (e.g., the importance of calendar time as a cost driver) and the underlying mathematical approach. This is not surprising considering that SLIM is based on the Putnam model [17] and JS-3 is based on the Jensen model [18], which, in turn, represents a modified Putnam model.

PCOC and WICOMO are both based on Boehm's Constructive Cost Model (COCOMO) [2]. PCOC implements Intermediate and Detailed COCOMO and WICOMO implements Detailed COCOMO. Both can be characterized as extremely "white box" in nature in that the underlying formulas and cost multipliers are conceptually straightforward and are visible to and easily changed by the user. Thus, both of these models can be readily tailored to specific organizations. WICOMO allows the user to structure software components into hierarchies with an arbitrary number of levels. Another noteworthy feature of WICOMO is its cost; it is by far the least expensive of the seven models.

PRICE S [16] is based on a cost-estimation methodology originally developed by RCA for estimating hardware costs (PRICE H). PRICE S has separate programs to

assist the user in estimating program size (PRICE SZ) and in predicting maintenance costs (PRICE SL).

SoftCost is based on a cost model developed by Tausworthe (Deep Space Network Software Cost Estimation Model) [19]. SoftCost allows the user to define a detailed work breakdown structure (WBS) or to use a default WBS that is compatible with MIL-STD-2167. The WBS can be combined with the estimates of effort and calendar time to produce detailed GANTT and PERT charts.

SPQR/20 is based on a cost model developed at ITT. In addition to cost, schedule, and staffing estimates, it makes a number of predictions related to product quality. These include the number of defects at various points in the life cycle, the number of test cases and test runs required, and the effectiveness of pre-test and test activities. SPQR/20 also predicts enhancement and maintenance activities.

### **C. VENDORS**

The name, address and telephone number of each of the vendors are listed in Exhibit 1.



Exhibit 1. MODEL VENDORS

<u>MODEL</u>	<u>VENDOR</u>
JS-3	Computer Economics, Inc. 4560 Admiralty Way Suite 109 Marina del Rey, CA 90292 (213) 827-7300
PCOC	Eclectic Systems P.O. Box 3461 Torrance, California 90510 (213) 618-1132
PRICE S	RCA PRICE Systems Route 38 Cherry Hill, NJ 08358 (609) 866-6583
SLIM	Quantitative Software Management, Inc. 1057 Waverley Way McLean, VA 22102 (703) 790-0055
SoftCost	Reifer Consultants, Inc. 25550 Hawthorne Boulevard Suite 208 Torrance, CA 90505 (213) 373-8728
SPQR/20 Quick Sizer	Software Productivity Research, Inc. 2067 Massachusetts Avenue Cambridge MA 02140 (617) 495-0120
WICOMO Tool	School of Information Technology Wang Institute of Graduate Studies Tyngsboro, MA 01879 (617) 649-9731

#### D. HARDWARE CONFIGURATION

Exhibit 2 summarizes the hardware requirements for each of the models. It is noteworthy that six of the seven models are available for the IBM PC (or compatibles). Additional details concerning hardware requirements are contained in Appendix B.

Exhibit 2. HARDWARE REQUIREMENTS

	JS-3	PCO	PRI	SLI	SOF	SPQ	WIC
IBM-PC (or compatibles)	X	X		X	X	X	X
WANG PC							X
PRIME 750 (PRIMOS)			X				X
VAX 780 (VMS)	X						X
VAX 780 (UNIX)							X
Apollo							X
Commercial Timesharing	X		X	X			

#### E. CONTRACTUAL ARRANGEMENTS

Exhibit 3 summarizes the contractual arrangements for each model. The models are fairly evenly divided in terms of their availability for purchase, lease, or through commercial timesharing. This information and information concerning costs was obtained through telephone conversations with each of the vendors in February and March 1986. In addition, the vendors were sent copies of a previous draft of this report for review and comment in May 1986. Additional contractual information is contained in Appendix B.

**Exhibit 3: CONTRACTUAL ARRANGEMENTS**

	PURCHASE	LEASE	TIMESHARE
JS-3		X	X
PCOC	X		
PRICE S		X	X
SLIM		X	X
SoftCost		X	
SPQR/20	X		
WICOMO	X		

**F. COSTS**

Exhibit 4 shows the DoD rates for each model. Appendix B contains additional details concerning user training costs (which are mandatory for some models), upgrades, and hotline assistance and information on non-DoD rates. It is clear from the table that there is wide variation in the cost of the models. In general, the more expensive models are associated with more sophisticated capabilities and with more extensive vendor support.

**Exhibit 4: COSTS (DoD Rates)**

	FIRST UNIT	ADDITIONAL UNITS
JS-3	\$9400 per year \$49.25 per hour connect time (time share version)	\$600 per year (PC version)
PCOC	\$850 one time charge	\$350 to \$175
PRICE S	\$75 per hour connect time (time share)	
SLIM	\$8000 per year	\$500 per year (PC version)
SoftCost	\$2500 for the first year / \$500 other years	Negotiable
SPQR/20	\$5000 one time charge	\$5000 to \$2500
WICOMO	\$200 one time charge	\$200

## **G. LIFE-CYCLE RANGE COVERED**

Exhibit 5 shows the range of life-cycle phases covered by each model. Two of the models cover development only and five encompass the operational phase as well. One difficulty in trying to compare models in terms of life-cycle coverage is that the phases are not defined in a standard way across models. The table describes each model in its own terminology.

Exhibit 5: LIFE-CYCLE RANGE COVERED

---

JS-3:	Requirements Definition up to Development Test and Evaluation plus fifteen years of Operational Support
PCOC:	Requirements Definition to Final Qualification Test plus five years of Operational Support
PRICE S:	Software Design through Test and Integration plus Operational Support of user-specified length
SLIM:	Feasibility Study to Full Operational Capability plus Operations and Maintenance
SoftCost:	System Specification (hardware and software) through System Test and Delivery
SPQR/20:	Planning through Integration/Test plus five years of Operational Support
WICOMO:	Product Design through Integration and Test

---

## **H. INPUT PARAMETERS**

In the next chapter, in which the discussion focuses on the evaluation criteria, the models are compared in terms of their capabilities and outputs. It is also of interest to examine the models from the perspective of the input parameters required, asking in particular:

- How many inputs are required by the different models?
- What categories of information do they address?

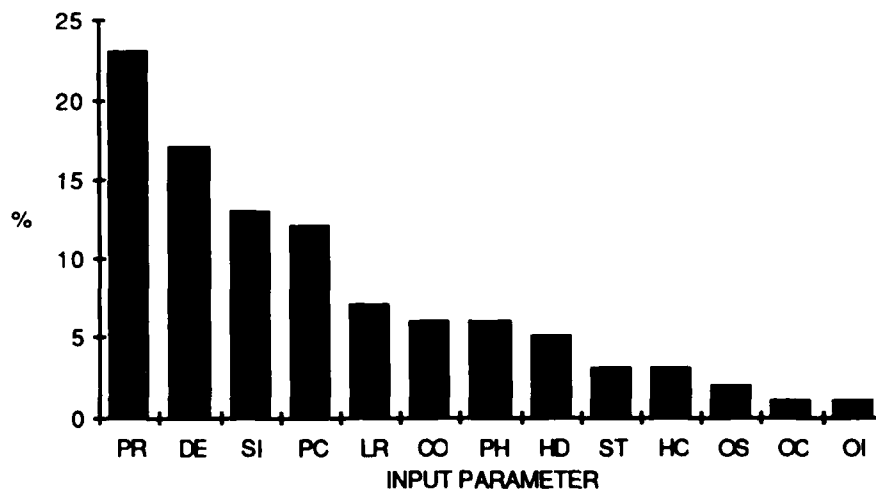
- Is there agreement among models about the types of inputs needed; in other words, is there consensus about what the relevant cost drivers are?
- Is this agreement at the level of specific inputs or is it more in terms of general categories of inputs?

Appendix C contains a list of all of the inputs required by the seven models. The number of inputs per model ranges from 21 (WICOMO) to 64 (PRICE S), with PCOC, SPQR/20 and WICOMO having approximately 25 and JS-3, PRICE S, SLIM, and SoftCost having approximately twice that number.

One possible strategy for validating contractor proposals would be to require contractors to submit, as part of their cost proposals, the values for a complete set of input parameters, which would allow the program office to run any or all the models. This is a feasible undertaking to the extent that the total number of input parameters is small. Across all models, there are a total of 285 input parameters, of which 190 are unique. The large number of unique parameters underscores the difficulty of implementing this strategy.

In order to look at the types of information encompassed by these parameters, they were classified into thirteen categories. (The assignment of individual parameters to categories is shown in Appendix C.) Exhibit 6 shows the distribution of parameters across categories. Of the 285, two-thirds fall into the categories of Product Complexity, Development Environment, Sizing Inputs, and Personnel Capabilities, suggesting a fair degree of consensus among models that these four categories represent important cost drivers. An examination of this distribution for each model separately shows a similar pattern in all cases with the exception of PRICE S. These distributions are presented in Appendix C.

Exhibit 6: DISTRIBUTION OF INPUT PARAMETERS BY INPUT CATEGORIES



Abbreviations denote the following categories:

PR:	Product Complexity
DE:	Development Environment
SI:	Sizing Inputs
PC:	Personnel Capabilities
LR:	Labor Rates
CO:	Constraints
PH:	Phase-Related Inputs
HD:	Historical Data for Calibration
ST:	Staffing
HC:	System Hardware Configuration
OS:	Operational Support
OC:	Other Costs
OI:	Other Inputs

The above comparison of input categories reflects consensus at a macrolevel. One can also look at the extent of agreement at a microlevel by calculating the proportion of specific parameters which are common to pairs of models. For example, if there are two models, one possessing 50 parameters, the other possessing 30 parameters, and 10 of the

parameters are common to each other then the proportion of overlap is calculated to be  $(10+10)/(50+30) = 20/80 = .25$ . The large number of unique input parameters (190 out of 285) already suggests a relatively low degree of consensus at the microlevel. Exhibit 7 presents these values for the seven models.

Exhibit 7: PROPORTION OF OVERLAP AMONG PAIRS OF MODELS

	JS-3	PCOC	PRICE S	SLIM	SoftCost	SPQR/20	WICOMO
JS-3	1.00	.33	.11	.27	.36	.13	.34
PCOC		1.00	.06	.21	.19	.07	.63
PRICE S			1.00	.20	.06	.00	.07
SLIM				1.00	.22	.08	.18
SoftCost					1.00	.16	.24
SPQR/20						1.00	.12
WICOMO							1.00

As can be seen from Exhibit 7, the amount of pairwise overlap is generally quite low, with a range from .00 to .63 and a mean value of .19. The highest proportion of shared input parameters is between PCOC and WICOMO, which is not surprising considering that both are based on the same underlying model (COCOMO). If one were to use more than one model for the purposes of a consistency check, one would probably choose ones that are maximally dissimilar in order to obtain as much independence in the estimates as possible. Thus, PCOC and WICOMO would not be an advisable pair for checking consistency with each other.

These analyses of input parameters point to a low degree of consensus among models in terms of the specific inputs required but a substantially higher degree in terms of general categories of input with Product Complexity, Sizing Inputs, Development Environment, and Personnel Capabilities accounting for the lion's share of the inputs.

The final section in this chapter compares the models in terms of the quality of user documentation.

## I. QUALITY OF THE USER DOCUMENTATION

Much of the information for this report was derived from the user's manuals. The clarity and overall quality of these manuals are important characteristics in their own right. The manuals were evaluated through the following five yes/no questions. (Questions 3, 4, and 5 were adapted from Adams and Halasz [28].)

Exhibit 8. QUALITY OF USER DOCUMENTATION

	JS-3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Does the user's manual stand-alone?	Y	N	Y	Y	N	Y	N
2. Is the manual written from the user's perspective?	Y	N	Y	Y	N	Y	Y
3. Are examples used?	Y	Y	Y	Y	Y	Y	Y
4. Are illustrations used in addition to text?	Y	Y	Y	Y	Y	N	Y
5. Is the type style attractive and varied?	Y	N	Y	Y	N	Y	N



#### IV. EVALUATION OF MODELS

The evaluation criteria, which appear in Chapter II, are presented as a series of questions. Responses to these questions for each of the seven models are in the form of detailed textual descriptions which are contained in Appendix D. The present chapter contains a summary of these descriptions in the form of ratings which range from 0 to 4, with a 0 indicating a complete absence of the relevant characteristic and 4 indicating a high degree of that characteristic. The models were rated on a relative basis: Except in cases which were clearly inappropriate, the model(s) with the greatest degree of a characteristic was assigned a 4 (appearing as a black dot in the following tables), any model(s) without a characteristic was assigned a 0 (appearing as a white dot), and the remaining models were assigned intermediate values (appearing as quarter, half and three-quarter dots). Appendix D should be referenced for a full description of the reasoning behind these ratings, which are based almost entirely on information contained in the user's manual for each model. A previous draft of this report was sent to each of the vendors for their review. As a result of their comments, several of the ratings were changed.

The criteria are not intended to provide a basis for ranking the models since not all criteria are equally important. Nevertheless, they do provide a basis for assessing the suitability of the alternative models for each of the usage categories. (As stated in Chapter I, empirical examination of these models was outside the scope of this study.)

While the evaluation criteria point to a number of differences among models, it is important to note that there are basic similarities as well. All estimate, at a minimum, the following quantities for each development phase (although the specific phases differ across models):






















- effort (manmonths)
- calendar time (months)
- staffing (average number of persons, rate at which people are added, or peak number of people).

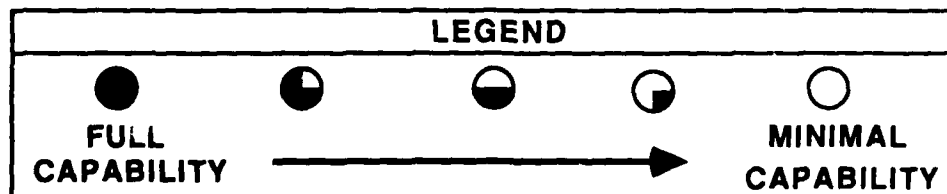
Exhibits 9 through 13 present a summary of the evaluation of the models.

The first category of criteria is concerned with how well the models assist in making basic investment decisions early in the life cycle. In the current study an average (mean) rating of 3 or 4 is viewed as an indication of the appropriateness of a model for a given usage category. As shown in Exhibit 9, three models support investment decisions early in the life cycle: JS-3, PRICE S, and SLIM. (In the exhibits, the model names are abbreviated by their first three letters.)

Exhibit 9. CRITERIA FOR USAGE CATEGORY #1

### ASSISTANCE IN MAKING BASIC INVESTMENT DECISIONS EARLY IN THE LIFE CYCLE

EVALUATION FACTOR	MODEL						
	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. CAN THE MODEL PROVIDE ESTIMATES ON THE BASIS OF MINIMAL INPUT?							
2. DOES THE MODEL PROVIDE AN INDICATION OF THE UNCERTAINTY OR RISK ASSOCIATED WITH ESTIMATES?							
3. IS ASSISTANCE PROVIDED IN ESTIMATING PROJECT SIZE?							

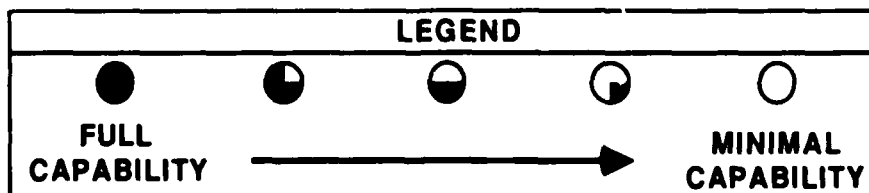


The second category of criteria is concerned with how well the models can help in validating contractor proposals. The results presented in Exhibit 10 suggest that JS-3, PRICE S, SLIM and SoftCost provide support for this purpose.

Exhibit 10. CRITERIA FOR USAGE CATEGORY #2

**ASSISTANCE IN VALIDATING CONTRACTOR PROPOSALS**

EVALUATION FACTOR	MODEL						
	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. DOES THE MODEL HAVE EASILY UNDERSTOOD INPUT PARAMETERS?	●	●	◐	◐	●	●	●
2. CAN THE MODEL BE CALIBRATED TO DIFFERENT ENVIRONMENTS?	○	●	◐	◐	●	◐	●
3. DOES THE MODEL SUPPORT A COMPARISON BETWEEN ORGANIZATIONS IN TERMS OF CAPABILITY?	●	◐	●	●	◐	◐	◐
4. DOES THE MODEL SUPPORT AN ANALYSIS OF COST-SCHEDULE TRADEOFFS?	●	●	●	●	◐	◐	◐
5. DOES THE MODEL INDICATE THE LIKELIHOOD OF COMPLETING WITHIN COST AND SCHEDULE?	●	○	◐	●	●	◐	○
6. DOES THE MODEL ALLOW CONSTRAINTS TO BE IMPOSED?	●	○	●	●	◐	○	○
7. IS PREVIOUSLY DEVELOPED SOFTWARE HANDLED?	●	◐	◐	◐	●	◐	○
8. DOES THE MODEL INCLUDE COSTS OTHER THAN LABOR?	○	○	●	◐	◐	○	○
9. DOES THE MODEL ACCOUNT FOR INFLATION?	●	●	●	●	○	○	○

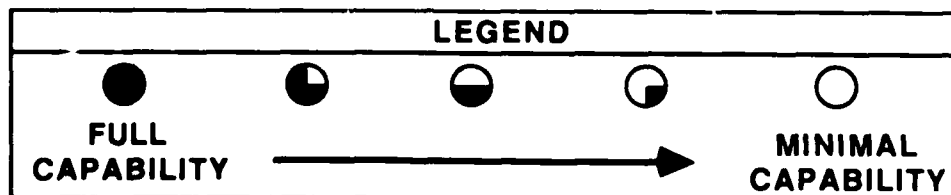


JS-3, PCOC, PRICE S, SLIM, and SoftCost provide support for day-to-day project management, the third category of criteria. Exhibit 11 summarizes the evaluation of the models against these criteria.

Exhibit 11. CRITERIA FOR USAGE CATEGORY #3

**SUPPORT FOR DAY-TO-DAY PROJECT MANAGEMENT**






















EVALUATION FACTOR	MODEL						
	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. DOES THE MODEL ALLOW THE USER TO DECOMPOSE THE SOFTWARE SYSTEM?	●	●	●	●	●	○	●
2. DOES THE MODEL PROVIDE WORK BREAKDOWN STRUCTURE?	◐	◐	◐	◐	●	◐	○
3. DOES THE MODEL SUPPORT THE ANALYSIS OF TASK DEPENDENCIES?	●	○	◐	○	●	◐	○
4. DOES THE MODEL PROVIDE SUPPORT FOR SENSITIVITY ANALYSES?	●	●	●	●	●	●	●
5. DOES THE MODEL PROVIDE A STAFFING PLAN?	◐	●	◐	●	◐	◐	◐
6. DOES THE MODEL PROVIDE A CASH-FLOW PLAN?	◐	◐	●	●	○	◐	◐
7. DOES THE MODEL PROVIDE A COMPARISON OF PLANNED EXPENDITURES VERSUS ACTUALS?	◐	◐	◐	◐	◐	○	○
8. DOES THE MODEL PROVIDE GRAPHICAL CAPABILITIES?	●	◐	◐	●	◐	◐	○

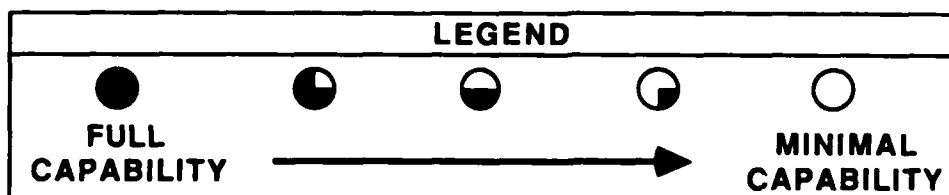


The fourth category of criteria is concerned with how well the models provide assistance in predicting enhancement and repair activities during the operation and maintenance phase. SLIM and SPQR/20 appear to be the best at meeting these criteria. The evaluation results are presented in Exhibit 12.

Exhibit 12. CRITERIA FOR USAGE CATEGORY #4

**ASSISTANCE IN PREDICTING ENHANCEMENT AND  
REPAIR ACTIVITIES DURING OPERATIONS AND MAINTENANCE**

EVALUATION FACTOR	MODEL						
	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. DOES THE MODEL PROVIDE ESTIMATES OF THE OPERATIONAL PHASE?							
2. DOES THE MODEL PREDICT NUMBER OR DENSITY OF DEFECTS?							
3. DOES THE MODEL PREDICT THE RUN-TIME BEHAVIOR OF THE SOFTWARE?							

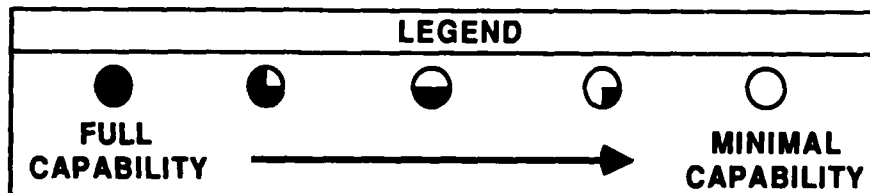


The final category of criteria focuses on how well the models support data analyses to identify major cost drivers and needed productivity improvements. The SoftCost model does a particularly good job in this area. The evaluation is shown in Exhibit 13.

Exhibit 13. CRITERIA FOR USAGE CATEGORY #5

**SUPPORT FOR ANALYSES TO IDENTIFY MAJOR COST DRIVERS  
AND NEEDED PRODUCTIVITY IMPROVEMENTS**

EVALUATION FACTOR	MODEL						
	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. DOES THE MODEL MAINTAIN A DATABASE OF INPUT VALUES?	●	●	●	●	●	○	●
2. DOES THE MODEL MAINTAIN THE RESULTING ESTIMATES?	●	◐	●	●	●	○	○
3. DOES THE MODEL MAINTAIN A DATABASE OF ACTUAL VALUES?	○	○	○	○	◐	○	○
4. DOES THE MODEL MAINTAIN A MULTIPLE-PROJECT DATABASE?	◐	○	○	◐	○	○	○
5. DOES THE MODEL ACCEPT EARLIER DATA FOR CALIBRATION?	◐	○	●	●	◐	○	○
6. CAN THE MODEL BE CHARACTERIZED AS A "WHITE BOX"?	◐	●	◐	◐	●	○	●



The strength of the one model not achieving a mean value of 3 or 4 in any of the usage categories (WICOMO) lies in its low cost, ease of calibration, and white-box nature, all of which are clearly important attributes.

## **V. SURVEYS OF USERS AND POTENTIAL USERS OF COST MODELS**

Three surveys were conducted during January and February of 1986 to assess the current state of practice in the DoD software community with regard to the use of software cost models. In addition to providing information about the extent of model use, these surveys provide insights into both factors that promote and factors that inhibit the use of models. Nearly all of the responses summarized here were obtained by telephone interviews.

The individuals contacted for the first survey were the points of contact for Mission Critical Computer Resource (MCCR) programs. They were asked to describe the use of software cost models within their programs or to provide a referral to someone who could. The second survey was conducted by contacting individuals who had taken an OASD-sponsored training class in the use of one automated cost model. The purpose of this survey was to determine whether they had used this education to apply software cost models at their workplaces. The respondents to the third survey were all known to be extensively involved with using software cost models. They all worked for the DoD or a software development contractor or were consultants to the DoD. The purpose of this survey was to obtain recommendations on how best to promote the use of models.

### **A. SURVEY 1: USE OF COST MODELS IN MCCR PROGRAMS**

A total of twenty MCCR programs were randomly selected from a list of 132 DoD programs having a significant software component [29]. The total hardware and software budgets for the programs selected range in size from \$4 million to \$250 million per year. The point of contact for each of the twenty programs was asked whether a software cost model had ever been used on the program. This could include use by the program office, by a software contractor, or by a third party. In many cases, the first office contacted for a program did not know whether any software cost models had ever been used for the program but could usually point to someone who (they thought) would know. There were three possible responses: 1) one or more models were used, 2) models were not

used, and 3) it was unknown whether models were used (or no useful information could be obtained).

Software cost models were reported to have been used by five of the twenty programs. In some cases, the models were employed by the program office and in other cases by the contractor or by an outside organization. In three of the five programs, models were employed only for an initial resource estimate. In the other two programs, models were also used to track expenditures against the initial estimate. One point of interest is that the negotiated price for one of the programs was considerably less than the cost estimated by the model, indicating a lack of confidence in the model estimates. However, that program is now about 50% over budget, a figure which is in fairly close agreement with the model's original projections.

For another five programs, software cost models were definitely not used, typically because cost models were not perceived as relevant to the particular program. At least one of these programs is currently experiencing a cost overrun in excess of 300%. The list of the reasons given for not attempting to use cost models for these programs is presented in Exhibit 14. The number of reasons exceeds the number of programs because two programs gave more than one reason.

#### Exhibit 14. REASONS FOR NOT USING MODELS

---

##### Models Not Seen As Appropriate

- The program is an upgrade.
- The program is a research and development effort.
- The program is thought to be unique.
- The program is an ongoing maintenance job.

##### Other Reasons

- Proposals to budget for modeling are always refused.
  - Only the total cost was estimated, not software alone.
  - Program is a fixed-price upgrade by the original developer.
-



It should be noted that several of the reasons given above do not preclude the use of software cost models. For example, several models can handle upgrades and maintenance, and one model (WICOMO) is available in an automated form for only \$200.

It was not possible to determine whether any cost model had been used in ten of the twenty programs, even after following leads and contacting personnel in several different offices. At the very least, this indicates an extreme lack of visibility of cost models in those projects, if any were used at all.

This survey points to a lack of widespread use of software cost models. Even though they were sometimes used for initial resource estimation (25% of the programs surveyed), only rarely were they used to help track or control a program (10% of the programs surveyed). The lack of general visibility of software models was apparent from the number of referrals which were frequently necessary in order to obtain information about any modeling activity as well as from the inability to determine whether models were used at all on ten of the twenty programs. The most common reason given for not using models was that no model would be appropriate. This points to either a need for further education in the use and availability of models or a need for more adaptive models.

#### **B. SURVEY 2: FORMER PARTICIPANTS IN OASD-SPONSORED TRAINING**

Of particular interest are the survey results from the group of people who had attended the OASD-sponsored training course because this represented a deliberate attempt to encourage cost-model use. Eighteen attendees of this course were contacted to find out whether the knowledge acquired during the course was applied when the attendee returned to his or her workplace. In those cases in which it was not, an effort was made to determine why not.

Of the eighteen attendees contacted, nine, or exactly half, have used cost models in some way since taking the course. However, five of these nine had already used cost models before taking the class. Of the remaining four, which was the group that subsequently tried the model demonstrated in the class, two have continued to use it, one discontinued its use in response to a budget cutback, and the fourth has changed jobs and does not know whether or not the model continues to be used in his previous location.

The fact that, in this sample, there were two or three additional converts to the use of cost models, indicates at least some success of the training class. In addition to the one case where model use was discontinued for budgetary reasons, it is useful to examine the other reasons for not using cost models, as told by the other nine participants.

The reasons given for non-use are presented below. Many of the respondents gave multiple reasons. Most (nearly three-quarters) of the responses obtained fell into two important categories. The first category involves a mismatch between the course content and the participant's a priori expectations about the training course, and the second category involves cases in which users returned to their workplace and found significant barriers to utilizing a cost model. A third category contains the remaining miscellaneous responses.

**CATEGORY #1: Mismatched Expectations about the Training Course (6 responses)**

- expected general education in cost models, not training in the use of one particular model
- claimed the course description was not accurate (two respondents)
- expected a way of estimating modifications, mostly small ones
- was looking for a planning tool not a cost model
- project was too small to take advantage of model

**CATEGORY #2: Barriers at the Workplace (12 responses)**

- management argued with results of first attempt at model use
- budget cuts prevented the lease or purchase of models
- management was not made aware of benefits
- respondent was at too low a level in the organization to require or even promote the use of cost models (four respondents)
- project schedules could not be controlled, as model required
- model was not made available back at workplace for follow-up and practice
- took course too late for the project at hand (two respondents)
- need never materialized, project never attempted

**CATEGORY #3: Miscellaneous (7 responses)**

- attended class only to evaluate it
- took only for background or enrichment in the field (five respondents)
- couldn't remember anything about the course

It is instructive to ask what could be done differently to encourage a higher rate of cost-model utilization among the course participants. From category #1, it is clear that a non-trivial proportion of the participants should not have been in the course simply because they were interested in something other than what the course offered. This problem can be solved relatively easily by providing more complete information about the course beforehand. (It seems, however, that the respondent who expected a "planning tool" may never have appreciated that the model being demonstrated in the class also assists in staffing and scheduling.) From category #2, the largest category, it is apparent that training alone is not sufficient. In a number of cases, the participants received little or no support from their home organizations to allow them to use the model. This suggests that any effective strategy to encourage cost-model use must take a much broader approach, particularly by encouraging the active support of management within the home organizations.

### C. SURVEY 3: KNOWN USERS OF COST MODELS

Eight respondents to this survey gave detailed comments on the use of cost models in their organizations. All were frequent and expert users of cost models, with one respondent indicating that he had produced more than 500 estimates. From the responses, the following four observations were judged the most significant.

#### 1. Mandate the use of models.

The difficulty and fear associated with using cost models for the first time within an organization should not be underestimated. This can prevent the adoption and use of even the most promising tool. The best way to promote the widespread use of cost models is simply to mandate their use, forcing people off "square one". Payback seems to occur within a year or two after initiating the use of models.

2. The advantages of using any model over using no model are so great that it almost doesn't matter what model is used; the differences among models are small by comparison.

Effort should be directed toward encouraging the use of a model. It is far less important which model is chosen. Considering the difficulty involved in getting people to use any model combined with the relatively minor differences among alternative models, users should begin with a model which is simple to learn and use.

3. Expect to invest time and effort to use models effectively.

Different models have different behaviors and sensitivities and it can require up to a year of use to learn to use a particular model to its best advantage. A software development organization should support a dedicated modeling team at a staff level to learn how to use the various cost models effectively within that organization.

4. More than one model should be used whenever possible.

The main reason for using more than one model is to cross check and to improve confidence in the results. Ideally, it makes more sense to use two models based upon different underlying algorithms than two based on the same mathematics. In addition, an organization or individual familiar with several models will come to understand the subtleties which make one model more appropriate than another in a given situation.

## VI. FINDINGS

The purpose of this study was two-fold:

- to identify and evaluate the currently available software cost models that could be used by DoD components, and
- to survey the DoD software community to assess the current state of practice with regard to the use of software cost models.

The results of the evaluation suggest that adequate models are available to support a wide variety of DoD software resource estimating needs. An average rating of 3 or 4 was viewed as an indication of the appropriateness of a model for a given usage category. By this criterion, it was concluded that

- JS-3, PRICE S, and SLIM support investment decisions early in the life cycle (Usage Category #1)
- JS-3, PRICE S, SLIM, and SoftCost provide assistance in validating contractor proposals (Usage Category #2)
- JS-3, PCOC, PRICE S, SLIM, and SoftCost provide support for day-to-day project management (Usage Category #3)
- SLIM and SPQR/20 support the operational phase (Usage Category #4)
- SoftCost supports analyses to identify major cost drivers and needed productivity improvements (Usage Category #5)

The strength of the one model not achieving an average value of 3 or 4 in any usage category (WICOMO) lies in its low cost (a \$200 one-time charge), its ease of calibration and its white-box nature, all of which are clearly important attributes.

The results of the surveys suggest that software cost-model use is not widespread. Even when a model is used, it tends to be employed on a one-shot basis rather than as an on-going management tracking tool. If cost models are to serve as effective planning and tracking tools, they must be given much greater visibility and support at much higher levels within DoD programs. Education for managers in the benefits and applicability of cost models is as important as education for the technically-oriented user. There were several cases where users were trained and willing to use models but where management failed to support this capability. In some cases, the resistance was passive, such as not

providing the necessary computer access, while in other cases management argued with the results of modeling or just rejected them.

**APPENDIX A**  
**REFERENCES**

## A. REFERENCES

- [1] Electronics Industry Association. "DoD Computing Activities and Program, 10-Year Market Forecast Issues, 1985-1995."
- [2] Boehm, B.W. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, 1981.
- [3] Aron, J.D. "Estimating Resources for Large Programming System." NATO Science Committee, Rome, Italy, October 1969.
- [4] Black, R.K.D., et al. "BCS Software Production Data." Final Technical Report, RADC-TR-77-116, Boeing Computer Services, Inc., March 1977.
- [5] Herd, J.R., et al. "Software Cost Estimation Study - Study Results." Final Technical Report, RADC-TR-77-220, Doty Associates, Inc., June 1977.
- [6] Bourdon, G.A. and Duquette, J.A. "A Computerized Model for Estimating Software Life Cycle Cost (Model Concept)." United States Air Force, April 1978.
- [7] Dodson, E.N., et al. "Advanced Cost Estimating and Synthesis Techniques for Avionics." Final Report, CR-2-461, General Research Corporation, May 1975.
- [8] Wolverton, R.W. "The Cost of Developing Large-Scale Software." IEEE Transactions on Computers, June 1974, pp. 615-636.
- [9] Boehm, B.W. and Wolverton, R.W. "Software Cost Modeling: Some Lessons Learned." Proceedings, Second Software Life-Cycle Management Workshop, U.S. Army Computer Systems Command, Atlantic, August 1978.
- [10] Buck, F., et al. "A Cost-By-Function Model for Avionics Computer Systems." NADC-SD-7088, March 1971.
- [11] Halstead, M.H. Elements of Software Science. New York: Elsevier, 1977.
- [12] Walston, C.E. and Felix, C.P. "A Method of Programming Measurement and Estimation." IBM Systems Journal, Vol. 16, No. 1, 1977, pp. 54-73.
- [13] Bailey, J.W. and Basili, V.R. "A Meta-Model for Software Development Resource Expenditures." Proceedings, Fifth International Conference on Software Engineering, IEEE/ACM/NBS, March 1981, pp. 107-116.
- [14] Parr, F.N. "An Alternative to the Rayleigh Curve Model for Software Development Effort." IEEE Transactions on Software Engineering, May 1980, pp. 291-296.
- [15] Nelson, E.A. "Management Handbook for the Estimation of Computer Programming Costs." Systems Development Corp., October, 1966.



- [16] Freiman, F.R. and Park, R.E. "PRICE Software Model-Version 3: An Overview." Proceedings, IEEE-PINY Workshop on Quantitative Software Models, October 1979, pp. 32-41.
- [17] Putnam, L.H. "A General Empirical Solution to the Macro Software Sizing and Estimating Problem." IEEE Transactions on Software Engineering, July 1978, pp. 345-361.
- [18] Jensen, R.W. "A Macrolevel Software Development Cost Estimation Methodology." Proceedings of Fourteenth Asilomar Conference on Circuits, Systems & Computers. New York: IEEE, 198 .
- [19] Tausworthe, R.C. "Deep Space Network Software Cost Estimation Model." Jet Propulsion Laboratory, JPL Publication 81-7, April 1981.
- [20] Thibodeau, R. "An Evaluation of Software Cost Estimating Models." General Research Corp., Rep. T10-2670, April 1981.
- [21] System-3 Featuring the JS-3: Software Estimation and Schedule Estimating Plus Decision Support Users Manual. Computer Economics Inc., June 1986.
- [22] PCOC Users Guide (Version 7.01), 1985.
- [23] PRICE S-3 Reference Manual (Revision II), October 1984.
- [24] SLIM User Manual (Draft Copy) for the IBM PC, July 1984.
- [25] SoftCost User's Manual, RCI-TN118 (V.5.1), Reifer Consultants, Inc., January 1986.
- [26] User's Guide, Software Productivity, Quality, and Reliability Estimator, SPQR/20 Quick Sizer, Version 1.0, September 1985.
- [27] Wang Institute Cost Model (WICOMO) Tool User's Manual. Software Product Report PUM1 Release 1.3-84, June 1984.
- [28] Adams, K.A. and Halasz, I.M. "25 Ways to Improve Software User Manuals." Proceedings of SOFTFAIR: A Conference on Software Development Tools, Techniques, and Alternatives. IEEE Computer Society, 1983.
- [29] Nash, S.H. "DoD Programs Containing a Significant Software Development Component (Spring 1984)." Institute for Defense Analyses, IDA Memorandum Report M-44, July 1985.

**APPENDIX B**  
**GENERAL INFORMATION**

## APPENDIX B. GENERAL INFORMATION

### A. HARDWARE CONFIGURATION

JS3: JS-3 runs on IBM PC's, XT's, AT's or compatibles with a minimum of 512K bytes of memory or a 448K Zenith 100 on a single floppy disk (although two are recommended). 640K and a high capacity or hard disk is required to run full graphics. JS-3 works with most popular video displays, in color or monochrome. The system requires PC DOS 2.0 (or MS DOS 2.0) or greater. An on-line version called "JS-DST" can be accessed via a time-sharing system with any office terminal and standard modem. JS-DST can also be installed on a VAX (under VMS) at a user's site. JS-DST was not reviewed for this study.

PCOC: PCOC runs on IBM PC's or compatibles (at least Zenith and Compaq are known to work) with a minimum of 192K bytes of memory. PCOC is delivered on a single 5-1/4 inch double-sided double-density floppy diskette and will support either a color or monochrome monitor.

PRICE S: PRICE S is available on a commercial time-sharing system and can be operated from an office terminal with a modem over standard telephone lines. It can also be installed on a PRIME 750 (under PRIMOS) at the user's site.

SLIM: SLIM runs on IBM PC's or compatibles with a minimum of 128K bytes of memory. One double-sided, double-density flexible disk drive, a color graphics monitor adapter board, and a CRT which supports graphics (preferably color) are required. SLIM is also available through a commercial time-sharing system. That version was not reviewed for this study.

SoftCost: SoftCost runs on IBM PC's, XT's, AT's or compatibles (such as Kaypro, Texas Instruments, Hewlett Packard, Eagle, Leading Edge, and others) with a minimum of 92K bytes of memory and one disk drive. SoftCost requires DOS 2.0 or greater.

SPQR/20: SPQR/20 runs on IBM PC's or compatibles with a minimum of 128K bytes of memory. One double-sided, double-density flexible disk drive is required.

WICOMO: WICOMO is available on tape for the VAX 780 (under UNIX and VMS), or the Prime 750 (under PRIMOS). It is also available on a single double-sided, double-density floppy diskette for the WANG PC, the Apollo, and the IBM PC.

## **B. CONTRACTUAL ARRANGEMENTS AND COSTS**

JS-3: JS-3 is available for lease only. The U.S. Government price is \$9400/year for the first unit and \$600/year per additional unit at the same site or division. The licensed software may be used on any compatible CPU within the site or division. A help line is available to users for no extra charge as are upgrades. In a special arrangement for government users (DoD and NASA), a dial-up to Wright Patterson AFB is available. Contact Lt. Paul Marsey (513) 255-6347 at Wright Patterson.

A three-day training course is strongly recommended at a cost of \$790/person when given at the vendor's (CEI) facility. A government rate of \$3800 (plus travel and living expenses for CEI personnel) allows for an unlimited number of students when given at the user's site.

PCOC: PCOC is available for purchase only. There is a sliding scale as follows: The price is \$850 per user for the first copy, \$350 each for the second through the tenth copy, and \$175 for the eleventh copy and on. The term "user" refers to a company and not a person or CPU. PCOC may be used by any employee of that company on any compatible CPU. One year of maintenance is included in the purchase price. PCOC upgrades are provided to users at a minimal replacement charge.

PRICE S: PRICE S runs on twin PRIME 750's at Morristown, New Jersey and is accessible through a commercial network. A government-wide contract covers NASA, NSA, FAA, as well as DoD. Government users can go through Aeronautical Systems Division (ASD, part of Systems Command) by contacting Lt. John Jones 513-255-6347 at Wright Patterson AFB. (Other agencies may have to make special arrangements.) These users see one bill each month for the connect time so it is cost effective to use PC-based terminals to upload and download results. The rate for this connect time is \$75/hour and is fixed until September 1988.

A week-long training course is mandatory and costs \$1125 per student for Government users. Refresher training is available for any student as are updates of manuals at no

additional charge. Technical assistance is available to users by phone or on site if necessary at no charge. Admission to PRICE seminars and symposia is also available at no additional charge.

Commercial rates for the package containing PRICE S, SL, and SZ (which assists in computing the machine instruction size input for S and SL) begin at \$38,500/year for one terminal connection at a time but unlimited use for the year. If simultaneous users are required, unlimited access costs an additional \$23,100. There are also charges for connect time (\$13/hour) and CPU time (\$0.055 per computer resource unit). This typically works out to \$6-7 for an average 15 minute run for both connect time and computer resource units. To use the program on the user's own PRIME computer, the above basic rates apply plus \$20,000 so a single user would cost \$58,500/year and an unlimited number of users costs \$81,600/year. The mandatory training costs \$1500 per student for commercial users.

SLIM: SLIM is available for lease only. The DoD rate is \$8000 per site per year plus \$500 for each additional CPU. A three-day training course is mandatory at a cost ranging from \$5000 to \$7000 when given on-site or for a cost of \$900 per student (with a discount available for volume) at the vendor's facility.

The commercial rate for SLIM is \$35,000 per year for the first site plus \$500 for each additional CPU. The commercial rate includes one on-site class at no extra charge. The commercial rate for each additional site is \$10,000 per year. There is a charge (\$5000-\$7000 for on-site or \$900 per student at the vendor's site) for the three-day course.

These rates include hot-line support at no extra charge.

SoftCost: SoftCost is available for a one-time lease charge of \$2500 per CPU. Training and consulting support are available for an additional charge. Included is one year of maintenance and updates (about three updates are issued per year). After the first year, the yearly maintenance and update agreement costs \$500 per year per CPU. Site, facility, or corporate licenses are negotiated on an individual basis.

SPQR/20: SPQR/20 is available for purchase only. There is a sliding scale as follows: The price is \$5000 per CPU for 1 to 3 units, \$4000 for 4 to 10 units, \$3000 for 11 to 20 units, and \$2500 for 21 units or more. This includes hot line support at no additional charge as well as maintenance for a 90-day period. After 90 days, a maintenance contract

is available for a fee. Users will receive their full purchase price when exchanging SPQR/20 for future upgrades.

WICOMO: WICOMO is available for purchase only at a cost of \$200 per CPU. This include the source code and three manuals. There is no maintenance or hot line support.

### **C. SIZE OF PROJECTS HANDLED AND OTHER CONSTRAINTS**

JS-3:       - Size: greater than 2,000 LOC  
              - Duration: up to five years Full Scale Implementation phase per CSCI  
              - Peak Staffing: five or more persons

PCOC:       - Size: up to 3,200,000 lines of code  
              - Duration: up to five years for development

PRICE S:   - None mentioned

SLIM:       - Size: 5,000 to 2,000,000 lines of code  
              - no other constraints are listed

SoftCost:   - no constraints are listed

SPQR/20:   - Size: 20 to 20,000,000

WICOMO:   - no constraints are listed

### **D. LIFE-CYCLE RANGE COVERED BY THE ESTIMATES**

Note: One problem in trying to compare cost models in terms of life-cycle coverage is that the phases are not defined in a standard way. The life-cycle phases covered by each model are described in terms of the model's own terminology.

JS-3: Predictions concerning development activities encompass the Requirements Definition Phase up to Development Test and Evaluation, that is, the point at which the software has been integrated with other software and hardware and has passed system tests. JS-3 also predicts effort, costs, and staffing for fifteen years of operational support which includes defect repairs, optimizations, and updates.

PCOC: Predictions concerning development activities encompass the Requirements Definition Phase to the Final Qualification Test. PCOC also predicts effort, cost, and staffing for five years of operational support.

PRICE S: Predictions concerning development activities encompass Software Design (including architectural as well as detailed design), Implementation, and Test and Integration. An additional program (PRICE SL) encompasses the operational phase, the number of years of which are specified by the user.

SLIM: The major portion of SLIM's analytical capabilities are focused on the development phases from the Preliminary Design Review (PDR) to Full Operational Capability. Effort, cost, and calendar-time projections are available for the Feasibility Study and Functional Design Phases as well as for the Operational and Maintenance Phase.

SoftCost: The model is applicable from system specification (hardware-software) through system test and delivery but can be run on subsets of this life cycle range. The user has an option of three different starting points to begin the WBS activities and the cost estimation computations.

SPQR/20: Predictions concerning development activities encompass Planning through Integration/Test. The development phases are not well defined nor are they associated with clear milestones. SPQR/20 also predicts effort, costs and staff size for maintenance (defect repair) and enhancements for a five-year period following delivery.

WICOMO: The predictions encompass Product (Preliminary) Design to Integration and Test. (Requirements definition and maintenance are not included.)

## **E. MILESTONES DEFINED**

- JS-3:
- System Requirements Review
  - Contract Award
  - Software Requirements Review (System Design Review)
  - Preliminary Design Review
  - Critical Design Review
  - Code and Unit Test
  - Final Qualification Test

- Software Development Test and Evaluation (Software End Product Acceptance)
- System Development Test and Evaluation Complete (System End Product Acceptance)

Note: All current Mil-Standards are supported and custom sets may be added.

- PCOC:
- Software Requirements Review
  - Preliminary Design Review
  - Critical Design Review
  - Code and Unit Test
  - Final Qualification Test

- PRICE S:
- None

- SLIM:
- Feasibility Study Review
  - Preliminary Design Review
  - Critical Design Review
  - First Code Complete
  - System Integration Test
  - User Oriented System Test
  - Initial Operational Capability
  - Full Operational Capability (95% reliability level)
  - 99% reliability level
  - 99.9% reliability level

- SoftCost:
- System Requirements Review
  - System Design Review
  - System Preliminary Design Review
  - System Critical Design Review
  - Software Requirements Review
  - Software Preliminary Design Review
  - Software Critical Design Review
  - Software Integration and Test
  - Software Test Readiness Review
  - Software Development Test and Evaluation



- Software Endproduct Acceptance Review
- System Acceptance Test

Note: SoftCost users may enter their own set of milestones into a file containing the work breakdown structure which is used to generate PERT and Gantt charts.

SPQR/20: - None

WICOMO: - None

## **F. SOFTWARE COST ELEMENTS ENCOMPASSED BY ESTIMATES**

JS-3:

- System Engineering
- Project Management
- Software Design
- Programming
- Quality Assurance
- Configuration Management
- Software Test
- Data Manipulation

PCOC:

- Requirements Analysis
- Product Design
- Programming
- Test Planning
- Verification and Validation
- Project Office
- Configuration Management/Quality Assurance
- Documentation

PRICE S:

- Systems Engineering
- Programming
- Configuration Control, Q/A
- Documentation
- Program Management

SLIM:

- Detailed Design
- Coding
- Test and Validation
- Documentation
- Management

SoftCost:

- Systems Engineering Support
- Software Project Management
- Software Development
- Software Quality Assurance

SPQR/20:

- Planning
- Requirements
- Analysis/Design
- Coding
- Integration/Test
- Documentation
- Management

WICOMO: None. All estimates are given in terms of phases only.

**APPENDIX C**  
**MODEL INPUT PARAMETERS**

## APPENDIX C. MODEL INPUT PARAMETERS

This Appendix presents a detailed listing of the model input parameters. The inputs have been categorized into seven categories as shown below.

R = Rating  
% = Percentage or proportion  
# = Count  
C = Category  
S = Character string  
Y/N = Yes or no  
E = Empirically derived

### Additional Notes:

- In a number of cases, the user is required (or has the option) of entering a range, consisting of minimum, most likely, and maximum values, rather than a single value. These cases include all inputs for JS-3, estimates of size for SLIM, and estimates of the percentage of source code to be delivered for SoftCost.

- PCOC allows the user to define up to three additional parameters and their weightings.

- Parameters listed for PRICE S include those for PRICE SL (Software Lifecycle package). Parameters specific to PRICE A (an activity analysis package) and PRICE SZ (a sizing package) are not included.

- In several cases, PRICE S provides alternative ways of entering a given piece of information. In particular, application complexity is a parameter which can be entered directly or calculated by PRICE S given a fairly complex set of inputs. Size is another parameter which can be entered in a variety of ways. In these tables, only the primary means of entry is shown.

- Several of the PRICE S parameters represent the combined effects of several factors and are empirically derived, making them difficult to summarize in a meaningful way in these tables. When the meaning is not obvious from the title of a parameter, it is enclosed in quotation marks. The reader is encouraged to consult the PRICE S Reference Manual for clarification of such cases. An example is the "Resource" parameter, which encompasses such items as skill levels, personnel experience, and overall organizational experience.

- In a few cases, parameter descriptions have been changed from the descriptions contained in the user's manuals in an effort to increase clarity.

Exhibit C-1. PRODUCT COMPLEXITY

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Storage optimization / Storage constraint / Target memory utilization	R	R		%	%		R
2. Application class complexity / Application type / Development mode	R	C	R	R			C
3. Timing optimization / Execution time constraint	%	R			%		R
4. Complexity of individual components	R	R					R
5. Portion of program which is real-time and/or multi-tasking	%		%	%	R		
6. Requirements volatility / Portion which (1) are stable, (2) will change slightly, and (3) will change drastically	R				%	R	
7. Required reliability/mean time to failure		R			#		R
8. Complexity of logical design					R	R	
9. Adaptation required to change from development to operational environment	R					R	

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
10. Difficulty of software and/or hardware interfaces	R					R	
11. Interfaces with other projects or organizations	R					R	
12. Data base size		R					R
13. Different I/O items to be generated per 1000 lines						R	
14. Overall complexity of program and data base architecture						R	
15. Percentage of the total task which will be easy						%	
16. Percentage of the total task which will be hard						%	
17. Code structure						R	
18. Data complexity						R	
19. "Complexity" (combination of complicating factors)			R				
20. Level of display interaction required for user interface	R						
21. Specification level	R						
22. Project novelty						R	
23. "Platform" (reflects requirements stemming from planned operational environment)			R				
24. Complexity of software requirements					R		
25. Customer/implementor organizational interface complexity					R		
26. Language complexity (number of years required to master)	R						
27. Software to be adapted to multiple environments					Y/N		
28. Classified security environment for computer					Y/N		
29. Amount of hardware under concurrent development					R		
30. Overall adverse constraints on program design					R		

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
31. Quality assurance level	R						
32. Test level	R						
33. Concept maturity	R						
34. Hardware utilization (timing and storage)			%				
35. Portion of program which involves operating systems application			%				
36. Portion of program which involves interactive operations			%				
37. Portion of program which involves on-line communications			%				
38. Portion of program which involves data storage and retrieval			%				
39. Portion of program which involves string manipulation			%				
40. Portion of program which involves mathematical operations			%				

Note: For #6, SoftCost requires three percentages.

#### Exhibit C-2. DEVELOPMENT ENVIRONMENT (TOOLS & METHODS)

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Turnaround / response time	#	R				R	R
2. Software development tools & environment reliability / Software tools	R	R			R		R

JS3 PCO PRI SLI SOF SPQ WIC

3. Maturity of system and support software / virtual machine volatility	R	R		R		R
4. Modern programming practices	R	R				R
5. Computer availability / Resource dedication	%		%	R		
6. Structured programming			R	R		
7. Design/code walkthroughs / Frequency of technical reviews to be held			R	R		
8. Top-down design / Top-down methodology			R	R		
9. Percent of work done at primary development site / Multiple site &/or organization development	R			R		
10. Computer accessibility / Access to development resources (in hours of travel time)	#			R		
11. Primary language / New code language			S		S	
12. Assembly language use			%	%		
13. Program librarian			R			
14. Online development			%			
15. New code language level					R	
16. Reusable code language					S	
17. Reusable code language level					R	
18. Secondary language			S			
19. Higher-order language use			%			
20. Fourth-generation language use			%			
21. Data base management system (DBMS) use			%			
22. Reports to be created by a report-writer utility			%			
23. Screens to be created by a screen-writer utility			%			



JS3 PCO PRI SLI SOF SPQ WIC

24. Machine processing power (scaling factor relative to IBM 370/155)				R		
25. Automated design support					R	
26. User documentation support					R	
27. Office facilities					R	
28. Virtual machine complexity	R					

Exhibit C-3. SIZING INPUTS

JS3 PCO PRI SLI SOF SPQ WIC

1. Size of code to be developed as completely new modules	#				#	#	#
2. Size of system or individual components (no distinction between new and existing software)		#	#	#			
3. Size of existing, to-be- inherited modules	#				#	#	
4. Size of code to be deleted from existing modules	#		#		#		
5. Size of code to be added to existing modules			#		#		
6. Size of code to be changed in other ways in existing modules	#				#		
7. Size of code remaining unchanged but to be retested / Reusable code size					#	#	
8. Portion of design that is new			%	%			
9. Portion of code that is new			%				
10. Portion of existing design modified		%					
11. Portion of existing code modified		%					

JS3 PCO PRI SLI SOF SPQ WIC

12. Integration and test effort required for modified software	%					
13. Lines of code to be deleted as entire modules					#	
14. Percentage of source code developed to be delivered					%	
15. Design effort required (compared with all new development)	%					
16. Implementation effort required (compared with all new development)	%					
17. Test effort required (compared with all new development)	%					
18. Source code reusability						R
19. Logically distinct input screens						#
20. Output types						#
21. Inquiry types						#
22. Data file accessed						#
23. Interfaces to other programs						#
24. Anticipated increase in system size during total operational phase				%		

Note: For all models except PRICE S, size is expressed as the number of lines of executable source code. For PRICE S, size is expressed as the number of deliverable, executable machine-level instructions.

Note: Parameters #19 through 23 are used in calculating function points.

# Exhibit C-4. PERSONNEL CAPABILITIES

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Team's experience (in years) with similar projects / Team's application experience		R		R	R	R	R
2. Staff's experience (in years) with operational computer(s) / Virtual machine experience	#	R		R	R		R
3. Staff's experience (in years) with programming language(s)	#	R		R	R		R
4. Analyst capability	%	R					R
5. Programmer capability	%	R					R
6. Overall team qualifications				R	R		
7. Customer experience in the application area					R		
8. "Resource" (measure of organizational performance)			E				
9. "Maintenance resource"			E				
10. "Enhancement resource"			E				
11. "Growth resource"			E				
12. Portion of designers who will be involved in implementation					R		
13. Experience of management team with similar projects				#			
14. Average staff experience					#		
15. Staff's experience with team concepts					R		
16. Expected user involvement in requirements definition					R		
17. Efficiency of implementing organization					R		
18. Analyst applications experience (in years)	#						

### Exhibit C-5. LABOR RATES

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Average labor rate	#			#		#	
2. Average annual inflation / escalation rate	%		%	%			
3. Monetary unit	S		S	S			
4. Design Phase			#				
5. Preliminary Design Phase							#
6. Detailed Design Phase							#
7. Code & Unit Test Phase			#				#
8. Integration & Test Phase			#				#
9. Labor rate for each labor category		#					
10. Management overhead to be applied to individual labor categories		%					
11. Cost per military personnel month	#						
12. Cost per civilian government personnel month	#						
13. Cost per contractor personnel month	#						

### Exhibit C-6. CONSTRAINTS

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Maximum development schedule	#	#		#			
2. Schedule pressure		R					R

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
3. Maximum development effort	#			#			
4. Maximum development cost			#	#			
5. Maximum peak manpower	#			#			
6. Minimum peak manpower				#			
7. Maximum Design Phase effort (manmonths/month or manhours/month)			#				
8. Maximum Implementation Phase effort			#				
9. Maximum Integration & Test Phase effort			#				
10. Maximum staffing rate (persons per year)	#						
11. Minimal probability of successful completion within schedule, staffing, and cost	%						
12. Minimal probability of successful completion within schedule					%		

#### Exhibit C-7. PHASE-RELATED INPUTS

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Effort expended in System Integration relative to Development	%		%				
2. Phase in which software work will begin					C		
3. Effort expended in Requirements Definition relative to effort expended in Development	%						
4. Distribution of effort across development phases		%	%				

JS3 PCO PRI SLI SOF SPQ WIC

5. Requirements Definition work  
completed prior to Contract  
Award

%

6. Distribution of schedule  
across development phases

%

7. Phase names

S

8. Distribution of activities  
across phases

%

9. Design start date

S

10. Design end date

S

11. Implementation start date

S

12. Implementation end date

S

13. Integration & Test start date

S

14. Integration & Test end date

S

15. Operational support start date

S

16. Operational support end date

S

#### Exhibit C-8. STAFFING

JS3 PCO PRI SLI SOF SPQ WIC

1. Manpower Buildup Index

E

2. Portion of peak staffing  
level available in-house

%

3. Portion of personnel  
available fulltime

%

4. Portion of personnel exempt  
from overtime

%

5. Average work week (hours)

#

6. Average work month (hours)

#

7. Average work year (days)

#

8. Distribution of labor categories  
across development activities

%

9. Labor category titles

S

### Exhibit C-9. SYSTEM HARDWARE CONFIGURATION

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Unique data storage and retrieval devices supported				#			
2. Total data storage and retrieval devices supported				#			
3. Unique on-line communication devices supported				#			
4. Total on-line communication devices supported				#			
5. Unique interactive devices supported				#			
6. Total interactive devices supported				#			
7. Unique real-time command and control devices				#			
8. Total real-time command and control devices				#			

### Exhibit C-10. OPERATIONAL SUPPORT

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Number of installations				#			
2. Average operational use relative to designed use				%			
3. Quality level of software system				%			
4. Level of enhancement to code during operational phase				%			
5. Estimated annual change traffic for first five years of operations				#			
6. Ratio of error rates of original code to added code				%			
7. Cost of the system modification				#			

### Exhibit C-11. HISTORICAL DATA FOR CALIBRATION

---

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Size of past project(s)			#	#			
2. Duration				#			
3. Development effort			#	#			
4. Application type			R	C			
5. Start date			S				
6. Completion date			S	S			
7. New design			%				
8. New code			%				
9. Inflation rate			%				

---

### Exhibit C-12. OTHER COSTS

---

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Cost per page of documentation				#			
2. Cost of purchased software			#				
3. Cost per CPU hour (\$)				#			
4. Cost multiplier to adjust for markups such as IR&D and fee			%				

---



Exhibit C-13. OTHER INPUTS

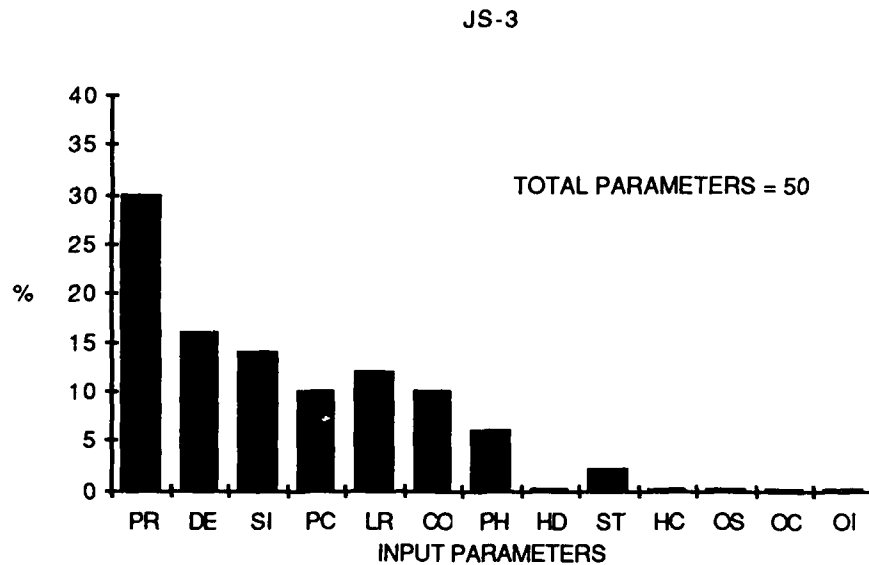
---

	JS3	PCO	PRI	SLI	SOF	SPQ	WIC
1. Expected economic life of system							#
2. Annual rate of return for investments							%

---

Exhibits C-14 through C-20 illustrate the distribution of each model's input parameters (by percent) across the thirteen input categories. The total number of input parameters is presented in the right-hand corner of each exhibit.

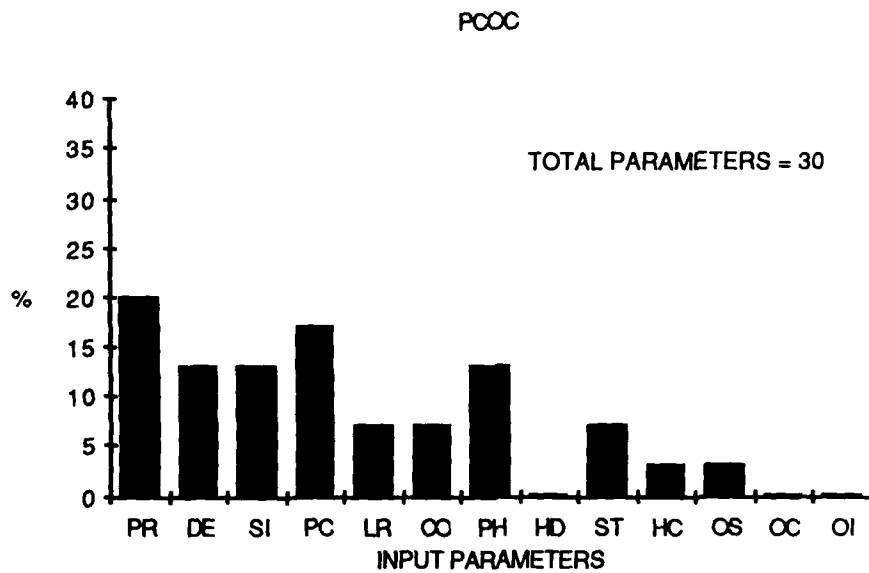
Exhibit C-14. JS-3 INPUT PARAMETER DISTRIBUTION



Abbreviations denote the following categories:

PR: Product Complexity  
DE: Development Environment  
SI: Sizing Inputs  
PC: Personnel Capabilities  
LR: Labor Rates  
CO: Constraints  
PH: Phase-Related Inputs  
HD: Historical Data for Calibration  
ST: Staffing  
HC: System Hardware Configuration  
OS: Operational Support  
OC: Other Cost  
OI: Other Inputs

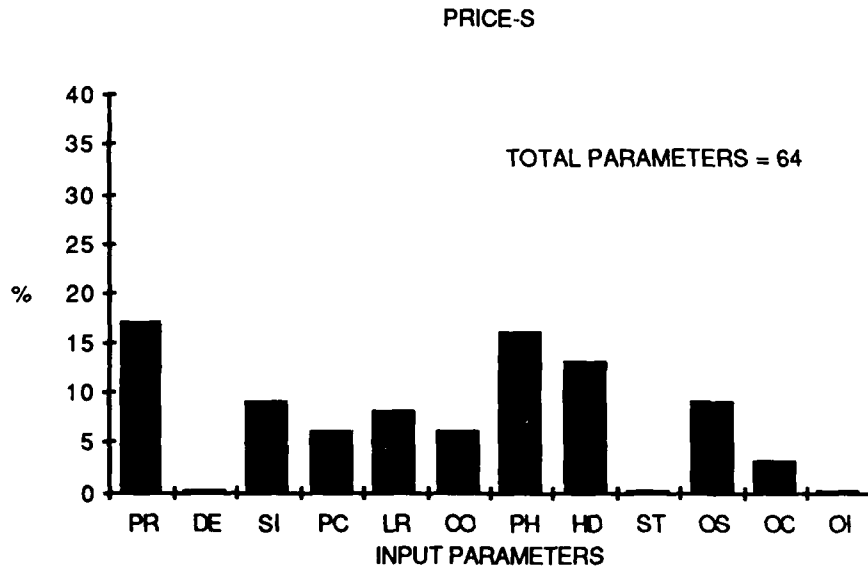
Exhibit C-15. PCOC INPUT PARAMETER DISTRIBUTION



Abbreviations denote the following categories:

PR: Product Complexity  
DE: Development Environment  
SI: Sizing Inputs  
PC: Personnel Capabilities  
LR: Labor Rates  
CO: Constraints  
PH: Phase-Related Inputs  
HD: Historical Data for Calibration  
ST: Staffing  
HC: System Hardware Configuration  
OS: Operational Support  
OC: Other Cost  
OI: Other Inputs

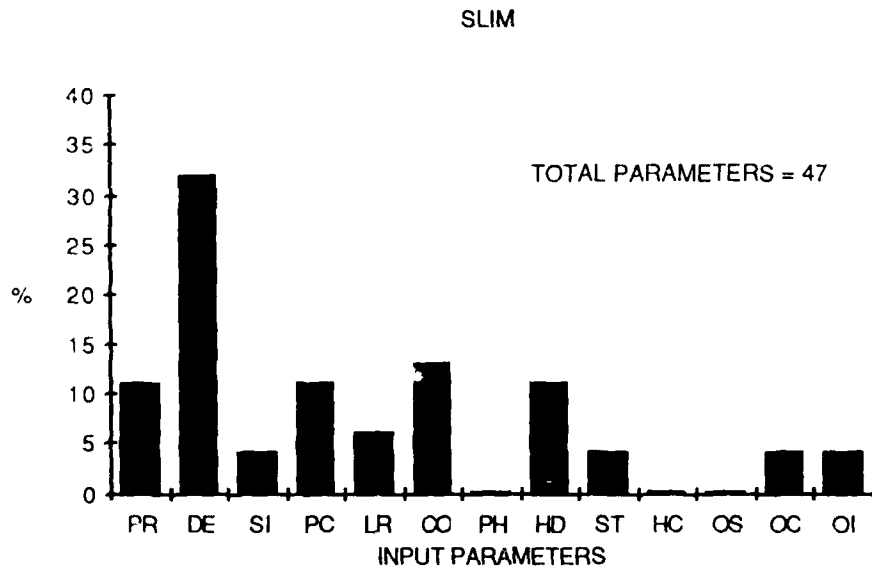
Exhibit C-16. PRICE-S INPUT PARAMETER DISTRIBUTION



Abbreviations denote the following categories:

PR:	Product Complexity
DE:	Development Environment
SI:	Sizing Inputs
PC:	Personnel Capabilities
LR:	Labor Rates
CO:	Constraints
PH:	Phase-Related Inputs
HD:	Historical Data for Calibration
ST:	Staffing
HC:	System Hardware Configuration
OS:	Operational Support
OC:	Other Cost
OI:	Other Inputs

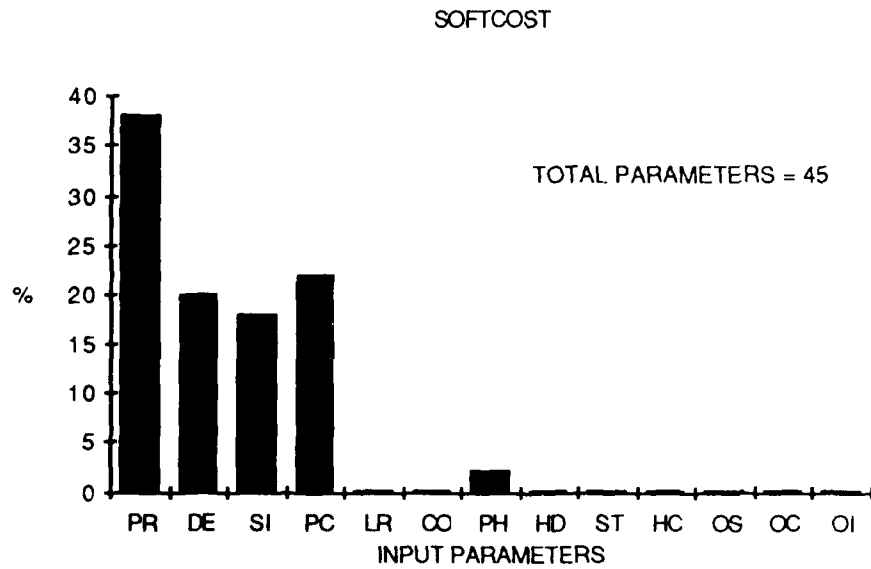
Exhibit C-17. SLIM INPUT PARAMETER DISTRIBUTION



Abbreviations denote the following categories:

PR:	Product Complexity
DE:	Development Environment
SI:	Sizing Inputs
PC:	Personnel Capabilities
LR:	Labor Rates
CO:	Constraints
PH:	Phase-Related Inputs
HD:	Historical Data for Calibration
ST:	Staffing
HC:	System Hardware Configuration
OS:	Operational Support
OC:	Other Cost
OI:	Other Inputs

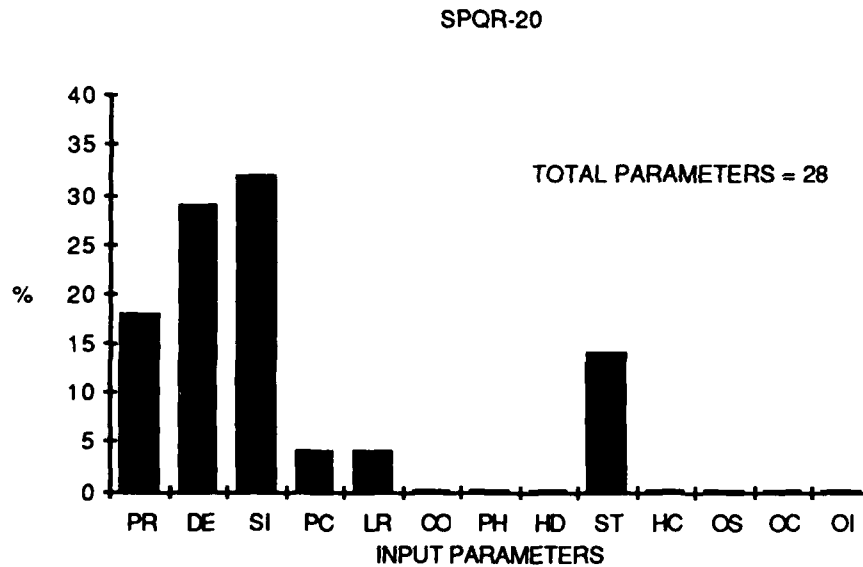
Exhibit C-18. SOFTCOST INPUT PARAMETER DISTRIBUTION



Abbreviations denote the following categories:

PR:	Product Complexity
DE:	Development Environment
SI:	Sizing Inputs
PC:	Personnel Capabilities
LR:	Labor Rates
CO:	Constraints
PH:	Phase-Related Inputs
HD:	Historical Data for Calibration
ST:	Staffing
HC:	System Hardware Configuration
OS:	Operational Support
OC:	Other Cost
OI:	Other Inputs

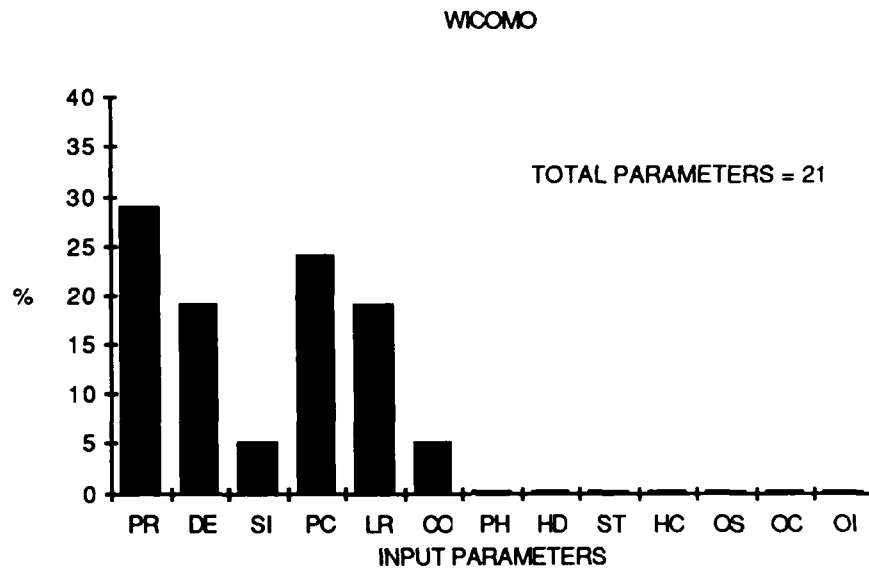
Exhibit C-19. SPQR/20 INPUT PARAMETER DISTRIBUTION



Abbreviations denote the following categories:

PR: Product Complexity  
DE: Development Environment  
SI: Sizing Inputs  
PC: Personnel Capabilities  
LR: Labor Rates  
CO: Constraints  
PH: Phase-Related Inputs  
HD: Historical Data for Calibration  
ST: Staffing  
HC: System Hardware Configuration  
OS: Operational Support  
OC: Other Cost  
OI: Other Inputs

Exhibit C-20. WICOMO INPUT PARAMETER DISTRIBUTION



Abbreviations denote the following categories:

PR:	Product Complexity
DE:	Development Environment
SI:	Sizing Inputs
PC:	Personnel Capabilities
LR:	Labor Rates
CO:	Constraints
PH:	Phase-Related Inputs
HD:	Historical Data for Calibration
ST:	Staffing
HC:	System Hardware Configuration
OS:	Operational Support
OC:	Other Cost
OI:	Other Inputs



**APPENDIX D**  
**DETAILED EVALUATION OF MODELS**

## **APPENDIX D. DETAILED EVALUATION OF MODELS**

**A. CRITERIA FOR USAGE CATEGORY #1:** Assistance in making basic investment decisions early in the life cycle

### **1.1 CAN THE MODEL PROVIDE ESTIMATES ON THE BASIS OF MINIMAL INPUT?**

**JS-3:** (Rating = 4) Yes. The user can treat the software system being estimated as a single component. Thus, detailed knowledge of the software structure is not needed. In addition, JS-3 provides nineteen different sets of default parameter values, allowing the user to enter, at a minimum, only size information in order to obtain the resulting estimates. These different default sets represent typical parameter values for specific application areas such as avionics, command and control, manned spacecraft, and so on. JS-3 also allows users to define their own default sets to represent typical projects within their organizations.

**PCOC:** (Rating = 4) Yes. The user can treat the software system being estimated as a single component. PCOC provides default values for all parameters except system size and labor rates.

**PRICE S:** (Rating = 3) Yes. The user can treat the software system being estimated as a single component but must provide a minimum of ten parameters. Most of these reflect information that should be known at an early point in the development if the software organization has developed similar products in the past.

**SLIM:** (Rating = 2) Partially. The user can treat the software system being estimated as a single component and enter size parameters (minimum and maximum values) for that component only. The user must provide input regarding a number of other parameters. A few default values are provided.

SoftCost: (Rating = 1) Partially. The user can treat the software product as a single entity. The user must, however, enter a detailed set of input parameters. No defaults are provided.

SPQR/20: (Rating = 2) Partially. SPQR/20 (always) treats the software product being estimated as a single entity. The user must enter a complete set of input parameters. Only a few default values are provided but most parameters reflect information that should be known at an early point in the life cycle.

WICOMO: (Rating = 4) Yes. The user can treat the software system being estimated as a single component. The minimum input involves entering non-zero values for system size and for the labor rate in each phase. WICOMO supplies nominal default values for all other input parameters.

## 1.2 DOES THE MODEL PROVIDE AN INDICATION OF THE UNCERTAINTY OR RISK ASSOCIATED WITH ESTIMATES?

JS-3: (Rating = 4) Yes. The concept of statistical probability or uncertainty is central to JS-3. This includes not only uncertainty of the JS-3 cost and schedule estimates but also uncertainty of the user's input values. For all input parameters, the user has the option of entering a range of values consisting of the minimum, most likely, and maximum values. From this range, JS-3 computes an expected value and a standard deviation. These, in turn, form the basis for a Monte Carlo simulation of the development activity. The basic estimates of effort and calendar time are given as expected values (at the 50% probability level). JS-3 also gives the probability of completing the development within specific months of calendar time and specific person months of effort.

PCOC: (Rating = 1) Partially. The user can obtain an indication of risk by executing separate runs with worst-case, most likely, and best-case values for input parameters and observing the effects on the effort and schedule estimates. Explicit probability values are not provided.

PRICE S: (Rating = 2) Partially. PRICE S provides the user with two types of sensitivity analyses to show the effects of variations in several important input parameters

on the resulting cost and schedule estimates. One analysis shows the effects of simultaneous variations in various project constraints (Complexity parameter) and in the characteristics of the resources applied (Resource parameter). The other analysis shows the effects of simultaneous variations in the size (Instruction parameter) and application mix of the product (Application parameter).

SLIM: (Rating = 4) Yes. The concept of statistical probability or uncertainty is central to SLIM. On the input side, the user enters estimates of size in terms of minimum, most likely, and maximum values. The resulting expected values and standard deviations are used by a Monte Carlo simulation of the development activity. The basic estimates of effort, cost, time, and staffing rate are given as expected values (at the 50% probability level) along with their standard deviations. SLIM also gives the probability that specific times and efforts will not be exceeded.

SoftCost: (Rating = 3) Yes. The estimates of effort and duration made by SoftCost are checked against a modified Putnam model. The latter, which relates effort and duration to program size, is used to indicate the size of program that can be developed given the estimated effort and duration. A confidence level is calculated on the basis of the discrepancy between the size estimated by the modified Putnam model and the size estimated by SoftCost. This confidence level reflects the probability of completing the project within both the estimated effort and duration.

SPQR/20: (Rating = 2) Yes. SPQR/20 predicts five "risk plateaus" ranging from "very high" to "very low". It also provides an explicit probability that the software system will fail completely, that is, will never be delivered. Finally, it points to likely problems such as high maintenance costs or low quality.

WICOMO: (Rating = 1) Partially. The user can obtain an indication of risk by executing separate runs with worst-case, most likely, and best-case values for input parameters and observing the effects on the effort and schedule estimates. Explicit probability values are not provided.

### 1.3 IS ASSISTANCE PROVIDED IN ESTIMATING PROJECT SIZE?

JS-3: (Rating = 3) Yes. Rather than attempting to estimate only one size value for a given component, the user inputs a range of estimated values representing the minimum, most likely and maximum number of lines of code (LOC) for each of four classes of software (new, existing, deleted and modified). From these input parameters, the model computes expected values and standard deviations (measures of uncertainty). Product size is expressed in terms of an "effective size" which accounts for the reduced effort involved in adapting existing code. The basic unit for LOC is a FORTRAN line or its equivalent. Pascal, COBOL, Ada, and most other high-order languages as well as assembly language are considered equivalent to FORTRAN for the purposes of estimating size. A non-equivalent language (e.g., a report-generator language) will require the user to manually make the conversion to the equivalent number of FORTRAN lines.

PCOC: (Rating = 0) No. The user inputs size. No special assistance is given.

PRICE S: (Rating = 4) Yes. The measure of size which drives the underlying cost-estimating algorithms is the number of delivered, executable, machine-level instructions (DEMI) rather than the number of lines of source code. Several alternatives for entering size are available as well as several forms of assistance. The user can enter an estimate of size in terms of DEMIs directly. Alternatively, the user can enter an estimate of the lines of source code along with a LOC-to-DEMI expansion factor. Another option is to enter the number of functional modules; PRICE S multiplies this number by 90 (or by a user-supplied multiplier) to arrive at the estimated DEMIs. There also exists an entire program called "PRICE SZ" for "PRICE Sizer" which calculates DEMIs (or source lines) based on several parameters including functional inputs and outputs as well as various characteristics of the development organization.

SLIM: (Rating = 2) Partially. The user enters the programming language(s) as well as minimum, most likely, and maximum values for lines of code. The model calculates the expected value and the standard deviation.

SoftCost: (Rating = 3) Yes. The user enters minimum, most likely, and maximum values for the size of each of seven different classes of code (e.g., completely new

modules, existing code, code to be changed). On the basis of these inputs, expected values and standard deviations are computed for each class. These are then used to derive an estimate of effective size which represents the equivalent number of new lines of code.

SPQR/20: (Rating = 3) Yes. The user has the option of entering size directly or of entering information needed to calculate function points from which a size estimate is derived.

WICOMO: (Rating = 0) No. The user inputs size. No special assistance is given.

## **B. CRITERIA FOR USAGE CATEGORY #2: Input into cost proposals**

### **2.1 DOES THE MODEL HAVE EASILY UNDERSTOOD INPUT PARAMETERS FOR WHICH DIFFERENT VALUES HAVE A READILY INTERPRETABLE MEANING?**

JS-3 : (Rating = 4) Yes.

PCOC: (Rating = 4) Yes.

PRICE S: (Rating = 2) Partially. Two of the major PRICE S parameters (Resource and Complexity) are not straightforward in that they represent the combined effects of multiple factors and are empirically derived. For example, the Resource parameter (to quote the PRICE S Reference Manual, p. III-6) " includes such items as skill levels, experience, productivity, efficiency, computer operating charges, labor and overhead rates "

SLIM: (Rating = 3) Yes. The possible exception of the Productivity Index and the Manpower Buildup Index which are empirically derived

SoftCost: (Rating = 4) Yes

SPQR 20: (Rating = 4) Yes

WICOMO: (Rating = 4) Yes.

## 2.2 CAN THE MODEL BE CALIBRATED TO DIFFERENT ENVIRONMENTS?

JS-3 (Rating = 0) No. The only form of "calibration" is based entirely on the parameter values entered by the user and not on an analysis of historical data. Nor can the user change JS-3's parameter weights or underlying formulas.

PCOC: (Rating = 4) Yes. All COCOMO constants are contained in a file which the user may edit. These constants include parameter weights as well as components of the basic formulas for each of the three types of estimating modes (organic, semi-detached, embedded). Hence, the user may change not only the weightings of the various cost-driver attributes but also the basic effort and schedule formulas. In addition, users may define their own mode along with effort and schedule formulas for that mode. Users may also define up to three additional cost drivers and their associated multipliers. Thus, it is physically easy for users to change the underlying constants to fit their organizations although PCOC does not help the user with the much more difficult conceptual task of deciding what the new set of constants should be.

PRICE S: (Rating = 3) Yes. Given historical data describing the size, schedule, and other characteristics of past projects, PRICE S calculates a Resource and a Complexity Index which reflect the performance of the software organization and an Application Index which reflects the characteristics of the software. These values can then be used for estimates concerning current and future projects. In addition, there are a number of ways in which users can customize PRICE S to reflect their particular ways of doing business.

SLIM: (Rating = 3) Yes. On the basis of size, effort, and calendar-time data from past projects, SLIM calculates two indices which serve to calibrate SLIM to the current project. One of these is the Productivity Index which reflects the efficiency of the developers and the complexity of past systems. The other is the Manpower Buildup Index which reflects the maximum effective buildup rate for the organization.

SoftCost: (Rating = 4) Yes. Users may edit parameter files. Users may also define their own, very detailed work breakdown structure which includes phases, activities within and across phases, precedence relationships, and distribution of time and effort across phases and activities.

SPQR/20: (Rating = 1) Partially. The user may examine and change the code production rate which is a measure of productivity that encompasses not only programmers but all project personnel. The change is in effect only for the current session rather than permanently.

WICOMO: (Rating = 4) Yes. All COCOMO constants are contained in a file which the user may edit. This includes parameter weights as well as components of the basic formulas. Hence, the user may change not only the weightings of the various cost-driver attributes but also the basic formulas relating size and effort as well as effort and schedule. Thus, it is physically easy for the user to change the underlying constants although WICOMO does not help the user with the more difficult conceptual task of deciding what the new set of constants should be.

### 2.3 DOES THE MODEL SUPPORT A COMPARISON BETWEEN TWO OR MORE ORGANIZATIONS IN TERMS OF CAPABILITY?

JS-3: (Rating = 4) Yes. The user may compare the Effective Technology Rating for each organization. This rating can be based on empirical data from past projects or user-provided ratings..

PCOC: (Rating = 2) Partially. The user may compare the values for input parameters reflecting the use of tools, modern development practices, experience and capability of the development team and so on.

PRICE S: (Rating = 4) Yes. The user may compare the Resource and Complexity Indices for each organization (provided the Application Indices are also comparable). This has the advantage of being derived from empirical data from past projects within the organizations of interest and, thus, reflects actual past performance.

SLIM: (Rating = 4) Yes. The user may compare the Productivity Index for each organization. This comparison is only valid when the projects upon which this index is based are of comparable complexity. This has the advantage of being derived from empirical data from past projects within the organizations of interest and, thus, reflects actual past performance.



SoftCost: (Rating = 2) Partially. The user may compare the values for input parameters reflecting the use of tools, modern development practices, experience and capability of the development team and so on.

SPQR/20: (Rating = 2) Partially. The user may compare the values for input parameters reflecting the use of tools, development-team experience and so on.

WICOMO: (Rating = 2) Partially. The user may compare the values for input parameters reflecting the use of tools, modern development practices, experience and capability of the development team and so on.

## 2.4 DOES THE MODEL SUPPORT AN ANALYSIS OF COST-SCHEDULE TRADEOFFS?

JS-3: (Rating = 4) Yes. JS-3 calculates costs, effort, and staffing rate based on a minimum calendar-time solution (50% probability of success). The user can enter variations in schedule, staffing rate, or effort as a constraint and JS-3 will calculate the remaining estimates based on this new constraint. A longer schedule than the minimum-time solution always results in reduced estimates of effort and cost.

PCOC: (Rating = 4) Yes. PCOC supports analysis of cost-schedule tradeoffs through the SCED input parameter which ranges from very low (schedule constrained to 75% of nominal) to very high (schedule stretched out to 160% of nominal or greater). Any deviation from nominal either shorter or longer results in higher estimated costs. In addition, the user can enter an explicit schedule constraint (in months) and observe the effect on estimated effort.

PRICES: (Rating = 4). Yes. PRICE S calculates a nominal or typical schedule. The user can enter shorter or longer schedules as a constraint. PRICE S will calculate costs and staffing level based on the new schedule. Any deviation (longer or shorter) from the nominal schedule results in higher estimated costs.

SLIM: (Rating = 4) Yes. SLIM calculates costs, effort, and peak staffing based on a minimum calendar-time solution (50% probability of success). The user can enter variations in schedule, cost or effort as a constraint and SLIM will calculate the remaining estimates based on this new constraint. A longer schedule than the minimum-time solution always results in reduced estimates of effort and cost.

SoftCost (Rating = 3) Yes. The user can hold one parameter constant and vary the other. The user can then observe the effect of these variations in terms of confidence levels. Lengthening the schedule for a given level of effort results in a higher estimated confidence level as does increasing the effort for a given schedule.

SPQR/20 (Rating = 2) Partially. The user can specify whether the goal of the analysis is to find a solution which minimizes costs, schedule, or whether a typical schedule and cost level is desired.

WICOMO: (Rating = 3) Yes. WICOMO supports analysis of cost-schedule tradeoffs through the SCED input parameter which ranges from very low (schedule constrained to 75% of nominal) to very high (schedule stretched out to 160% of nominal or greater). Any deviation from nominal either shorter or longer results in higher estimated costs.

## 2.5 DOES THE MODEL INDICATE THE LIKELIHOOD OF COMPLETING THE SOFTWARE WITHIN THE PROPOSED COST AND SCHEDULE?

JS-3 : (Rating = 4) Yes. Probability distributions are given for various efforts and times.

PCOC: (Rating = 0) No.

PRICE S: (Rating = 1) Partially. PRICE S displays an error message if a schedule or effort constraint is "over restrictive".

SLIM: (Rating = 4) Yes. Probability distributions are given for various efforts, costs, and times.

SoftCost: (Rating = 4) Yes. Probabilities are given.

SPQR/20: (Rating = 2) Partially. Probability of complete failure is given.

WICOMO: (Rating = 0) No.

## 2.6 DOES THE MODEL ALLOW DEVELOPMENTAL CONSTRAINTS TO BE IMPOSED?

JS-3 : (Rating = 4) Yes. Constraints can be imposed on maximum effort, maximum schedule, and maximum staffing rate.

PCOC: (Rating = 0) No.

PRICE S: (Rating = 4) Yes. Constraints can be imposed on desired schedule, maximum effort, and maximum peak manpower.

SLIM: (Rating = 4) Yes. Constraints can be imposed on maximum effort, maximum schedule, maximum peak manpower, and minimum peak manpower.

SoftCost: (Rating = 3) Yes. The schedule can be entered as a constraint and SoftCost will calculate the confidence levels associated with variations in effort. Conversely, effort can be entered as a constraint and SoftCost will calculate the confidence levels associated with variations in schedule.

SPQR/20: (Rating = 0) No.

WICOMO: (Rating = 0) No.

## 2.7 IS PREVIOUSLY DEVELOPED SOFTWARE HANDLED?

JS-3 : (Rating = 4) Yes. JS-3 very explicitly treats previously developed software. JS-3 takes into account whether a component represents a completely new development or whether it represents a modification of an existing component. In the latter case, JS-3 adds in additional overhead for lines of code being added, producing an "effective" size as one output which then drives the cost and schedule predictions. Other things being

equal, 50K new lines will require less effort and less time than the development of 50K lines embedded within an existing component.

PCOC: (Rating = 2) Yes. The overhead involved in modifying existing software is translated into additional lines of code. This overhead is based on the user's estimates of the percentage of the existing design and code which must be modified as well as the overhead involved in integrating and testing the modified software.

PRICE S: (Rating = 3) Yes. The overhead involved in modifying existing software is translated into an increased estimate of product size. PRICE S also explicitly adds in the cost of purchased software as well as the effort required to integrate both purchased and furnished software.

SLIM: (Rating = 1) Yes. However, relatively minor consideration is given to previously developed software. The user enters the percent of the detailed design ("algorithms and logic design") that is completely new.

SoftCost: (Rating = 4) Yes. SoftCost very explicitly treats previously developed software. In addition to modules consisting of entirely new code, SoftCost makes distinctions between six classes of existing code. The user gives estimates of the size of each class. These classes include (1) total size of existing code requiring modification, (2) size of code deleted from existing modules, (3) size of code added to existing modules, (4) size of code to be changed in other ways in existing modules, (5) size of code to be deleted as entire modules, and (6) size of code not modified but requiring re-testing. These various classes are assigned different weights by SoftCost in calculating the effective size of the entire system.

SPQR/20: (Rating = 3) Yes. There are several input parameters containing information about the amount of re-use of existing code and about the characteristics of that code.

WICOMO: (Rating = 0) No. (This contrasts with the COCOMO model which takes into account the overhead involved in modifying existing code.)

## 2.8 DOES THE MODEL INCLUDE COSTS OTHER THAN LABOR?

JS-3 : (Rating = 0) No.

PCOC: (Rating = 0) No.

PRICE S: (Rating = 4) Yes. PRICE S includes purchased software. Also, PRICE S makes specific allowance for overhead costs such as G&A, IR&D, and so on.

SLIM: (Rating = 3) Yes. SLIM includes documentation (in number of pages and cost) and CPU usage (in hours and cost).

SoftCost: (Rating = 3) Yes. SoftCost includes documentation (in number of pages and cost) and CPU usage (in hours and cost).

SPQR/20: (Rating = 0) No.

WICOMO: (Rating = 0) No.

## 2.9 DOES THE MODEL ACCOUNT FOR INFLATION?

JS-3: (Rating = 4) Yes. The user enters an average anticipated inflation rate over the duration of the project. The user may also enter an inflation rate for each year of operational support.

PCOC: (Rating = 4) Yes. The user enters an anticipated inflation rate for each year of the project.

PRICE S: (Rating = 4) Yes. PRICE S provides default values for inflation or the user can enter an inflation rate.

SLIM: (Rating = 4) Yes. The user enters an average anticipated inflation over the duration of the project.

SoftCost: (Rating = 0) No.

SPQR/20: (Rating = 0) No.

WICOMO: (Rating = 0) No.

**C. CRITERIA FOR USAGE CATEGORY #3: Support for day-to-day project management**

**3.1 DOES THE MODEL ALLOW THE USER TO DECOMPOSE THE SOFTWARE SYSTEM INTO SMALLER COMPONENTS?**

JS-3: (Rating = 4) Yes. The user may decompose a project into tasks, elements, and units. One or more units can be combined to form an element, one or more elements form a task, and one or more tasks form a project. Size and complexity information are entered at the level of units or elements. Most input parameters are entered at the level of a task. A few parameters (e.g., inflation rate) apply to the project as a whole.

PCOC: (Rating = 4) Yes. The user may define a two-level hierarchy with a maximum of sixteen top-level units and fifteen subunits for each unit. The COCOMO cost drivers can be defined at the unit or subunit level and are defined independently for each (sub)unit.

PRICE S: (Rating = 4) Yes. PRICE S allows the user to decompose the software into components and to specify the integration difficulty of each component. Costs and schedule are estimated separately for each component. The overall costs for system integration and test are also provided.

SLIM: (Rating = 4) Yes. The user may define up to one-hundred different components. For each component, the user enters size information. All other input parameters and all outputs are associated with the software system as a whole and not with the individual components.

SoftCost: (Rating = 4) Yes. The user may decompose a software system into subsystems, the number of which is limited only by disk space. All input parameters are entered for each subsystem.

SPQR/20: (Rating = 0) No.

WICOMO: (Rating = 4) Yes. The user may define any number of components at any number of levels. With a few exceptions, input parameters may be associated at any level. A lower-level component automatically inherits the value of a parameter from its parent. The user may then change that inherited value. Only the parameter values at the lowest levels are used in the actual estimates. Thus, parameters are defined at higher levels for the convenience that is derived from the inheritance characteristics of the hierarchy. WICOMO provides effort and cost estimates for any given component at any level of the hierarchy. Schedule and staffing-level estimates are given only for the top-level component (that is, for the software system as a whole).

### 3.2 DOES THE MODEL PROVIDE THE EQUIVALENT OF A WORK BREAKDOWN STRUCTURE?

JS-3: (Rating = 3) Yes. Estimates of cost or effort are broken down by phase into the activities of system engineering, project management, design, programming, quality assurance, configuration management, testing, and data manipulation (documentation). This breakdown is provided at the level of individual tasks, collections of tasks (called "groups"), or an entire project.

PCOC: (Rating = 1) Partially. PCOC shows the (COCOMO-provided or user-supplied) percentage of effort within each development phase that is devoted to the activities of requirements analysis, product design, programming, test planning, verification and validation, project office, configuration management/quality assurance, and documentation. PCOC does not, however, present a breakdown of estimated effort, costs, or staffing by activity. This must be calculated manually.

PRICE S: (Rating = 2) Yes. Estimates of cost or effort are broken down into the activities of system engineering, programming, configuration control-Q/A, documentation, and program management. PRICE S provides monthly as well as cumulative costs or effort for each activity.

SLIM: (Rating = 2) Yes. The estimated total effort is broken down into the activities of detailed design, coding, integration, test and validation, documentation, and management.

SLIM provides monthly staffing levels for each activity as well as monthly effort and cost expenditures.

SoftCost: (Rating = 4) Yes. SoftCost allows the user to define a detailed work breakdown structure which includes precedence information as well as information about the distribution of effort and time across activities. Given this work breakdown structure along with the estimates of effort and duration, SoftCost can generate detailed Gantt and PERT charts. The default provided by SoftCost is a work breakdown structure compatible with MIL-STD-2167.

SPQR/20: (Rating = 1) Partially. Overall estimates of effort, cost, and staff level are broken down into the activities of planning, requirements, analysis/design, coding, integration/test, documentation, and management.

WICOMO: (Rating = 0) No. Its estimates are broken down only by phase and not by any more detailed activities within and across phases.

### 3.3 DOES THE MODEL SUPPORT THE ANALYSIS OF TASK DEPENDENCIES?

JS-3: (Rating = 4) Yes. Software components (or "elements") are combined to make up a group work assignment (or "task"). JS-3 allows the user to try out different combinations of elements to determine the optimum task structure in terms of its effects on the overall project effort, cost, schedule, and staffing levels. JS-3 also determines which task in a collection of tasks requires the longest schedule. The schedule for all other tasks is then extended in order to minimize costs without adversely impacting the project schedule.

PCOC: (Rating = 0) No.

PRICE S: (Rating = 1) No. However, PRICE S does assume some overlap between phases. Shortening or lengthening the nominal schedule is assumed to result in inefficiencies in applying people across phases and, hence, in increased costs.

SLIM: (Rating = 0) No.



SoftCost: (Rating = 4) Yes. The work breakdown structure, which can be user-defined or provided by default, contains precedence information to allow the identification of critical paths and dependencies among activities. For activities in which there is leeway in that a larger window of time is available than needed, the earliest and latest beginning and ending points are labeled.

SPQR/20: (Rating = 1) No. SPQR/20 does provide two schedule estimates, one of which assumes no overlap among activities such as requirements definition and design and among design and code and the other of which assumes some overlap.

WICOMO: (Rating = 0) No.

#### 3.4 DOES THE MODEL PROVIDE SUPPORT FOR SENSITIVITY ("WHAT-IF") ANALYSES?

JS-3 : (Rating = 4) Yes. It's straightforward to change only one or a few parameters and rerun the analyses.

PCOC: (Rating = 4) Yes. It's straightforward to change only one or a few parameters and rerun the analyses.

PRICE S: (Rating = 4) Yes. In addition to two types of built-in sensitivity analyses (described under Question 1.2), it is straightforward to change only one or two parameters and recalculate the costs, schedule and manpower estimates.

SLIM: (Rating = 4) Yes. SLIM provides an initial set of effort, cost, time, and peak-staffing estimates based on a minimum-time solution. The user may then carry out a whole series of "what-if" analyses to examine the effects of various constraints as well as variations in a number of parameters.

SPQR/20: (Rating = 4) Yes. The user can easily modify one or more input parameters and rerun the analysis. The results of two analyses can be displayed side-by-side to facilitate comparison.

SoftCost: (Rating = 4) Yes. The user can easily modify one or more input parameters and rerun the analysis. Also, the user can hold the estimated effort constant and vary the

duration or hold duration constant and vary effort and observe the effect on the confidence level.

WICOMO: (Rating = 4) Yes. The user can easily modify one or more input parameters and rerun the analysis.

### 3.5 DOES THE MODEL PROVIDE A STAFFING PLAN THAT IS BROKEN DOWN INTO FREQUENT INTERVALS, SUCH AS MONTHLY?

JS-3: (Rating = 3) Yes. JS-3 provides a monthly staffing plan (shown relative to project milestones) broken down into Development Staff (those directly producing design and code) and Project Staff (other personnel such as managers, testers, quality assurance personnel, etc.).

PCOC: (Rating = 4) Yes. PCOC presents a monthly staffing plan, broken down into as many as sixteen different labor categories. Histograms are provided for the total labor profile as well as for any individual labor category.

PRICE S: (Rating = 3) Yes. PRICE S presents a monthly breakdown of effort for each of five project activities (system engineering, programming, configuration control-Q/A, documentation, integration and test).

SLIM: (Rating = 4) Yes. SLIM provides monthly, quarterly, and yearly staffing plans. These are presented in tabular form as well as in the form of a histogram showing the staffing levels relative to project milestones. A monthly breakdown of effort is also given for each of six project activities (detailed design, coding, integration, documentation, management, and test and validation).

SoftCost: (Rating = 1) No. SoftCost presents duration, effort, and start- and end-dates for up to 999 activities contained in the work breakdown structure. The user could manually generate a monthly staffing plan on the basis of this information.

SPQR/20: (Rating = 1) No. The average staff size is given for each of seven project activities (planning, requirements, analysis/design, coding, integration/test, documentation, management) but no further breakdown is given.

AD-A178 400

A DESCRIPTIVE EVALUATION OF AUTOMATED SOFTWARE  
COST ESTIMATION MODELS (U) INSTITUTE FOR DEFENSE  
ANALYSES ALEXANDRIA VA E R BAILEY ET AL. OCT 86

2/2

UNCLASSIFIED

IDA-P-1979 NDA903-84-C-0031

F/G 9/2

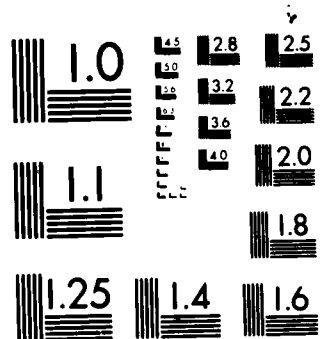
NL

END

DATE

FILED

4-87



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS 1963-A

**WICOMO: (Rating = 2) Yes. WICOMO gives a monthly breakdown of the current and cumulative effort for each phase.**

### **3.6 DOES THE MODEL PROVIDE A CASH-FLOW PLAN?**

**JS-3: (Rating = 2) Partially. JS-3 provides monthly cumulative costs but not monthly expenditures, JS-3 also shows the phase-by-phase cost for each of seven labor categories.**

**PCOC: (Rating = 3) Yes. PCOC presents a monthly cash-flow plan.**

**PRICE S: (Rating = 4) Yes. PRICE S presents a monthly breakdown of costs for each of five project activities (systems engineering, programming, configuration control-Q/A, documentation, integration and test).**

**SLIM: (Rating = 4) Yes. SLIM presents a cash-flow plan in the form of a table or as a histogram showing the monthly cash flow relative to project milestones. SLIM also presents the total cost for each of eight project activities (feasibility study, functional design, detailed design, coding, integration, documentation, management, and test and validation).**

**SoftCost: (Rating = 0) No.**

**SPQR/20: (Rating = 1) No. The cost is given for each of seven project activities (planning, requirements, analysis/design, coding, integration/test, documentation, management) but no further breakdown is given.**

**WICOMO: (Rating = 3) Yes. WICOMO gives a monthly breakdown of the current and cumulative cost for each phase.**

### 3.7 DOES THE MODEL PROVIDE A COMPARISON OF PLANNED EXPENDITURES AND MILESTONE-COMPLETION DATES VERSUS ACTUALS?

JS-3: (Rating = 2) Partially. Outputs can be stored for later reference but any comparison with actuals must be made manually or through a user-provided tool. Outputs are stored in a format to facilitate processing by other tools.

PCOC: (Rating = 2) Partially. Some outputs can be stored for later reference but any comparison with actuals must be made manually or through a user-provided tool. Input and output values are stored in a format to facilitate processing by other tools.

PRICE S: (Rating = 2) Partially. Outputs can be saved for later reference and analysis but any comparison with actuals must be made manually or through a user-provided tool. The user can create a machine-readable file to facilitate processing by other tools.

SLIM: (Rating = 2) Partially. Outputs can be stored for later reference but any comparison with actuals must be made manually or through a user-provided tool. Outputs are stored in a format to facilitate processing by other tools.

SoftCost: (Rating = 3) Partially. Outputs can be saved for later reference but any comparison with actuals must be made manually or through a user-provided tool. Gantt charts can be updated with actual completion dates for activities contained in the work breakdown structure.

SPQR/20: (Rating = 0) No.

WICOMO: (Rating = 0) No.

### 3.8 DOES THE MODEL PROVIDE GRAPHICAL CAPABILITIES?

JS-3: (Rating = 4) Yes. A number of different color (or black and white) displays and printouts are available in the form of histograms and continuous distributions.

PCOC: (Rating = 2) Yes. The graphics are character-oriented.

PRICE S: (Rating = 2) Yes. The graphics are character-oriented.

SLIM: (Rating = 4) Yes. Many different types of color (or black and white) graphical displays and printouts are available including histograms, continuous distributions, Gantt charts, and pie charts.

SoftCost: (Rating = 2) Yes. The graphics are character-oriented.

SPQR/20: (Rating = 2) Yes. The graphics are character-oriented.

WICOMO: (Rating = 0) No.

**D. CRITERIA FOR USAGE CATEGORY #4: Assistance in predicting enhancement and repair activities during operations and maintenance**

**4.1 DOES THE MODEL PROVIDE POST-DELIVERY ESTIMATES OF THE OPERATIONAL PHASE, INCLUDING OPTIMIZATIONS, ENHANCEMENTS, DEFECT CORRECTIONS?**

JS-3: (Rating = 3) Yes. JS-3 provides yearly effort and costs for fifteen years of operational support. Two effort estimates are provided: one for the development staff which involves those directly involved in design and programming and a second for the total staff which includes the development staff plus those involved in management, configuration control, etc.

PCOC: (Rating = 3) Yes. The user must estimate the "annual change traffic" which represents the percentage of the to-be-maintained code that will be modified or added. This estimate is entered for each of five years. The same basic estimates of effort, cost, and staffing are available as for development. (The schedule is partitioned by year rather than by development phase.)

PRICE S: (Rating = 4) Yes, these are addressed in detail. As part of the PRICE model, there is a program called PRICE SL (for Software Life-cycle) which specifically encompasses the operational phase. PRICE SL predicts the costs of Maintenance (defect-repair), Enhancement (performance-improvement), and Growth (additional functionality).

These predictions are broken down into the activity categories of system engineering, programming, configuration control-Q/A, documentation, and program management. For each activity, the estimates are presented yearly and as totals; they cover any number of years specified by the user and are based on a detailed characterization of the maintenance organization, the software product and its operational use (e.g., the number of installations).

SLIM: (Rating = 3) Yes. SLIM provides monthly, quarterly, or yearly staffing and cash-flow plans which are intended to cover modifications, enhancements, and defect-correction activity throughout the operational life of the software system.

SoftCost: (Rating = 2) Partially. SoftCost specifically considers the overhead involved in modifying existing code. As long as the enhancement activities during maintenance occur in a structured manner (as is assumed with development), SoftCost should be a valuable source of assistance. SoftCost does not provide separate estimates of the operational phase; rather this would have to be treated as a development activity. SoftCost does not specifically consider defect correction.

SPQR/20: (Rating = 3) Yes. SPQR/20 estimates the effort, cost, and staff size for enhancements and defect corrections for a five-year period following delivery of a software system.

WICOMO: (Rating = 0) No.

#### 4.2 DOES THE MODEL PREDICT THE NUMBER OR DENSITY OF DEFECTS?

JS-3 : (Rating = 0) No.

PCOC: (Rating = 0) No.

PRICE S: (Rating = 0) No.

SLIM: (Rating = 4) Yes. SLIM predicts both the total number and their density (defects per KLOC).

SoftCost: (Rating = 0) No.



SPQR/20: (Rating = 4) Yes. SPQR/20 predicts several quantities related to the number of defects and the effectiveness of testing.

WICOMO: (Rating = 0) No.

#### 4.3 DOES THE MODEL PREDICT THE RUN-TIME BEHAVIOR OF THE SOFTWARE USING A MEASURE SUCH AS MEAN TIME TO FAILURE (MTTF)?

JS-3 : (Rating = 0) No.

PCOC: (Rating = 0) No.

PRICE S: (Rating = 0) No.

SLIM: (Rating = 4) Yes. MTTF is projected by month until a reliability of 99.9% has been achieved (that is, 99.9% of the original defects have been found and corrected).

SoftCost: (Rating = 0) No.

SPQR/20: (Rating = 4) Yes. MTTF at delivery is predicted as well as MTTF at stabilization (which is defined as the post-delivery point at which the system becomes stable enough to be used productively).

WICOMO: (Rating = 0) No.

#### E. CRITERIA FOR USAGE CATEGORY #5: Support for analyses to identify major cost drivers and needed productivity improvements

##### 5.1 DOES THE MODEL MAINTAIN A DATABASE OF INPUT VALUES?

JS-3 : (Rating = 4) Yes.

PCOC: (Rating = 4) Yes.

PRICE S: (Rating = 4) Yes.

SLIM: (Rating = 4) Yes.

SoftCost: (Rating = 4) Yes.

SPQR/20: (Rating = 0) No.

WICOMO: (Rating = 4) Yes.

## 5.2 DOES THE MODEL MAINTAIN A DATABASE OF THE RESULTING ESTIMATES?

JS-3: (Rating = 4) Yes. Outputs are saved in a format so as to be readily processed by other tools.

PCOC: (Rating = 3) Yes. Some outputs are saved and are formatted so as to be readily processed by other tools.

PRICE S: (Rating = 4) Yes. Outputs are saved in a format so as to be readily processed by other tools.

SLIM: (Rating = 4) Yes. Outputs can be saved in a format so as to be readily processed by other tools.

SoftCost: (Rating = 4) Yes. Outputs can be saved in a format so as to be readily processed by other tools.

SPQR/20: (Rating = 0) No.

WICOMO: (Rating = 0) No.

## 5.3 DOES THE MODEL MAINTAIN A DATABASE OF ACTUAL VALUES?

JS-3: (Rating = 0) No.

PCOC: (Rating = 0) No.

PRICE S: (Rating = 0) No.

**SLIM: (Rating = 0) No.**

**SoftCost: (Rating = 3) Yes.** Actual completion dates for activities within the work breakdown structure can be entered and saved.

**SPQR/20: (Rating = 0) No.**

**WICOMO: (Rating = 0) No.**

**5.4 DOES THE MODEL MAINTAIN A MULTIPLE-PROJECT DATABASE WHICH CAN BE USEFULLY ACCESSED BY THE USER TO PROVIDE COMPARISONS AND VIEWS OF IMPROVEMENTS OVER TIME?**

**JS-3: (Rating = 1) Partially.** JS-3 can compare two projects in terms of their input parameters and will show those parameters that differ.

**PCOC: (Rating = 0) No.**

**PRICE S: (Rating = 0) No.**

**SLIM: (Rating = 1) Partially.** SLIM maintains a database of software projects which are used to compare current estimates for normalcy. A much more sophisticated comparison capability has been provided by this vendor via an additional product.

**SoftCost: (Rating = 0) No.**

**SPQR/20: (Rating = 0) No.**

**WICOMO: (Rating = 0) No.**

**5.5 DOES THE MODEL ACCEPT EARLIER DATA FOR THE PURPOSES OF CALIBRATION OR COMPARISON?**

**JS-3: (Rating = 2) Yes.** The user can enter size, schedule and effort data from earlier projects and JS-3 will compute an "effective technology rating" which reflects the use of tools, development practices, personnel capabilities and various design constraints. The

effective technology rating can also be computed from user-provided inputs about these characteristics.

PCOC: (Rating = 0) No.

PRICE S: (Rating = 4) Yes. These data are used to calculate the Resource, Complexity, and Application Indices which, in turn, are used to calibrate PRICE S to the current environment.

SLIM: (Rating = 4) Yes. These data are used to calculate the Productivity Index and the Manpower Buildup Index which, in turn, are used to calibrate SLIM to the current environment.

SoftCost: (Rating = 3) Yes. In addition to a user's manual, SoftCost users are provided with a Calibration Guide which shows how the data from past projects can be used (along with subjective judgment) to calculate a productivity adjustment constant which, in turn, is used to calibrate SoftCost to the current environment.

SPQR/20: (Rating = 0) No.

WICOMO: (Rating = 0) No.

## 5.6 CAN THE MODEL BE CHARACTERIZED AS "WHITE BOX"?

JS-3: (Rating = 1) Partially. All major equations are published. However, the user cannot change these equations or the parameter weights.

PCOC: (Rating = 4) Yes. All parameter weightings as well as the coefficients and exponents of the basic effort and schedule formulas are available for examination and editing by the user. PCOC provides a number of ways in which users can tailor the model to their own needs. For example, users can define an additional cost-estimation mode complete with its own basic effort and schedule formulas beyond the three that are defined by COCOMO (organic, semi-detached, embedded). They can also rename the development phases and define up to sixteen different labor categories.

**PRICE S:** (Rating = 1) Partially. While much of the computation is hidden, the user is able to customize PRICE S to a large extent. As examples, the user can enter multipliers for any of the fifteen development cost estimates (five activities for each of three phases) to compensate for any systematic over- or under-estimation observed on past projects, there are a number of defaults values that can be changed, and the user can specify the shape of the manpower curve for each phase.

**SLIM:** (Rating = 1) Partially. All major equations are published. However, the user cannot change these equations or the parameter weights.

**SoftCost:** (Rating = 4) Yes. All parameter weightings as well as the work breakdown structure and several other files which are used by the model are available for examination and editing by the user.

**SPQR/20:** (Rating = 0) No.

**WICOMO:** (Rating = 4) Yes. All parameter weightings as well as the coefficients and exponents of the basic effort and schedule formulas are available for examination and editing by the user.

ATE  
LMED  
-8